

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

**Návrh a implementace front-end pro
konfiguraci zařízení měření tlaku**

**Pressure Measuring Device Front-end
Design and Implementation**

Zadání diplomové práce

Student:

Bc. Radim Kula

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2612T041 Řídicí a informační systémy

Téma:

Návrh a implementace front-end pro konfiguraci zařízení měření tlaku
Pressure Measuring Device Front-end Design and Implementation

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Seznámení se s použitými technologiemi. (HTML, JavaScript, PHP, CSS, Bootstrap, AJAX, JQuery, MySQL, Raspberry PI)
2. Seznámení se s měřením tlaku.
3. Návrh a implementace technického řešení front-endu.
4. Testování vytvořeného řešení.
5. Závěrečná zhodnocení.

Seznam doporučené odborné literatury:

- [1] DUCKETT, Jon. *Web Design with HTML, CSS, JavaScript and jQuery Set*. United States: John Wiley, 2014. ISBN 978-1785287053.
- [2] MORETO, Silvio. *Bootstrap 4 By Example: Community experience distilled*. Packt Publishing, 2016. ISBN 978-1785287053.
- [3] NIXON, Robin. *Learning PHP, MySQL & JavaScript: With JQuery, CSS & HTML5: Learning PHP, MYSQL, Javascript, CSS & HTML5*. 5, ilustrované vydání. O'Reilly Media, Incorporated. ISBN 978-1491978917.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jaromír Konečný, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019

doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2019



.....

Rád bych na prvním místě poděkoval vedoucímu mé diplomové práce, panu Ing. Jaromíru Konečnému, Ph.D. za obrovskou trpělivost, rady a připomínky při tvorbě diplomové práce. Také bych rád poděkoval svým kamarádům, rodině za podporu nejen při tvoření této práce, ale i při průběhu celého studia.

Abstrakt

Tato práce řeší návrh a implementaci webové aplikace frond-endu pro zařízení měřící tlak. Výstupem práce je nástroj pro nastavování zařízení. Mimo to také pro zobrazení naměřených dat pomocí webového prohlížeče. Pro implementaci je využit jazyk PHP společně s CSS, JavaScriptem a databází MySQL.

Klíčová slova: HTML, JavaScript, PHP, CSS, Bootstrap, Ajax, jQuery, MySQL, WEB, TLAK

Abstract

This thesis deals with a front-end development and implementation of the web application for a pressure measurement device. An outcome is a tool for setting the pressure measurement device but too visualization measured data on the web page. For implementation is using programing language PHP together with CSS, JavaScript and for databases using MySQL.

Key Words: HTML, JavaScript, PHP, CSS, Bootstrap, Ajax, jQuery, MySQL, WEB, PRES-SURE

Obsah

Seznam použitých zkratk a symbolů	7
Seznam obrázků	9
Seznam tabulek	10
Seznam výpisů zdrojového kódu	11
1 Úvod	12
2 Seznámení se s použitými technologiemi	13
2.1 HTML	13
2.2 CSS (Cascading Style Sheets)	15
2.3 JavaScript	17
2.4 PHP	19
2.5 Bootstrap	22
2.6 jQuery	24
2.7 AJAX	26
2.8 MySQL	27
2.9 Raspberry Pi	28
3 Seznámení se s měřením tlaku	31
3.1 Základní definice tlaku a jednotky	32
3.2 Senzory používané při měření tlaku	35
4 Návrh a implementace technického řešení front-endu	39
4.1 Návrh webové aplikace	40
4.2 Implementace jednotlivých částí webové aplikace	42
4.3 Návrh MySQL databáze	49
5 Testování vytvořeného řešení	54
5.1 Technické parametry vytvořené webové stránky	54
5.2 Ukázky ošetření nežádoucích stavů	54
6 Závěr	57
Literatura	58

Seznam použitých zkratk a symbolů

API	– Application Programming Interface – rozhraní pro programování aplikací
ARM	– Architektura procesorů
CGI	– Common Gateway Interface – protokol pro propojení externích aplikací s webovým serverem
CSS	– Cascading Style Sheets – kaskádové styly
DOM	– Document Object Model – objektový model dokumentu
F	– Síla
GB	– Gigabyte – označení jednotky množství informace
GPIO	– General-purpose input/output – vstupně/výstupní piny
GPL	– General Public License – licence pro svobodný software
GPU	– Graphics processing unit – grafický procesor
HDMI	– High-Definition Multi-media Interface – port pro přenos videa a audia
HTML	– Hyper Text Markup Language – název značkovacího jazyka
I2C	– Inter-Integrated Circuit – počítačová sériová sběrnice
LAN	– Local Area Network – lokální síť
MHz, GHz	– Jednotka frekvence – megahertz, gigahertz
PEAR	– PHP Extension and Application Repository – repozitář kódu pro PHP
PECL	– PHP Extension Community Library – modul PHP
PHP	– Hypertext Preprocessor, dříve Personal Home Page – skriptovací programovací jazyk
POSIX	– Portable Operating System Interface – označení standardu zejména pro unixové operační systémy
PWM	– Pulse Width Modulation – diskrétní pulzně šířková modulace pro přenos analogového signálu
R	– Univerzální tlaková konstanta
RAM	– Random-Access-Memory – paměť s přímým přístupem umožňující čtení i zápis
S	– Plocha
SoC	– System on a chip – počítač integrovaný v jediném integrovaném obvodu
SPI	– Serial Peripheral Interface – sériové periferní rozhraní
T	– Termodynamická teplota
UNIX	– Ochranná známka operačního systému

USB	– Universal Serial Bus – univerzální sériová sběrnice
V	– Objem
W3C	– World Wide Web Consortium – mezinárodní konsorcium
Wi-Fi	– Wireless – bezdrátová komunikace
XHTML	– eXtensible Hyper Text Markup Language – název značkovacího jazyka
XML	– eXtensible Markup Language – obecný značkovací jazyk
g	– Gravitační zrychlení
h	– Výška
hPa, kPa, MPa, Pa	– Pascal – jednotky tlaku
k_u	– Napěťová konstanta snímače
l	– Délka
n	– Látkové množství v molech
p_c	– Celkový tlak
p_d	– Dynamický tlak
p_k	– Kinetický tlak
s	– Stlačitelnost kapaliny
w	– Rychlost proudění tekutiny
ρ	– Hustota tekutiny
ε_0	– Permitivita vakua
ε_r	– Permitivita použitého piezoelektrického materiálu
$^{\circ}\text{C}$	– Jednotka teploty

Seznam obrázků

1	Rozdíl mezi HTML 4 a HTML 5.[2]	14
2	Ukázka propojení logických urovní CSS a HTML.	17
3	Znázornění základních kamenů moderní webové stránky.	18
4	Logo Bootstrapu.[9]	22
5	Ukázka aplikování frameworku Bootstrap.	24
6	Znázornění pojmů jQuery.	25
7	Příklad, jak funguje Ajax.	26
8	Vývoj MySQL.	27
9	Raspberry Pi 3 B+.[16]	29
10	Blokové schéma Raspberry Pi.	29
11	Rozhraní GPIO na Raspberry Pi.[15]	30
12	Bourdonová trubice.[19]	32
13	Měření tlaků.	34
14	Znázorňuje drátový tenzometr, kde 1 je odporový drát, 2 je deformační podklad, 3 jsou konektory tenzometru. [18]	36
15	Příklad znázorňující princip funkce piezoelektrického snímače tlaku.[21]	37
16	Diferenciální kapacitní senzor s oddělovací kapalinou. [20]	38
17	Kompletní měřicí řetězec	39
18	Koncept webové aplikace	40
19	Koncept obsahu webové stránky pinu a nodu.	41
20	Koncept obsahu webové stránky manuálního ovládání.	42
21	Koncept obsahu webové stránky naměřená data.	43
22	Okno volby souboru pro importování konfiguračních údajů do databáze.	44
23	Případy užití webové aplikace.	48
24	Případy užití konfigurace pinu/nodu.	49
25	Případy užití manuálního ovládání.	49
26	Případy užití naměřených dat.	50
27	Celková struktura databáze.	51
28	Ukázka okna změn a volby možnosti zápisu do databáze a hardwaru.	56

Seznam tabulek

1	(measure) číslo měření.	51
2	(measure_data) data měření.	52
3	(pin) parametry pinu.	52
4	(node) parametry nodu.	52
5	(command) příkazy.	53
6	Popis jednotlivých zkratk v textovém řetězci DATA.	53
7	Časy načítání webové stránky po stisku Ctrl + R	55
8	Časy načítání webové stránky po stisku Ctrl + F5	55

Seznam výpisů zdrojového kódu

1	Základní struktura webového dokumentu	15
2	Příklad inline zápisu CSS kódu	16
3	Příklad interního zápisu CSS kódu	16
4	Příklad externího zápisu CSS kódu	16
5	Příklad externího zápisu JavaScript kódu	19
6	Ukázka PHP kódu	22
7	Ukázka jQuery kódu	26
8	Ukázka části kódu navigační lišty webové aplikace	47
9	Ukázka souboru <code>define_lang.php</code>	47
10	Část výpisu kódu ze souboru <code>lang_cs.php</code>	47
11	Ošetření stejné volby.	55

1 Úvod

V dnešní době s rozvojem průmyslu 4.0, kdy je kladen důraz na sběr informací a snahu vše automatizovat, byl vytvořen požadavek na sestavení zařízení měřící tlak uvnitř lisovacího zařízení. U takového projektu bude potřeba kromě hardwarové části také vyvinout grafické rozhraní pro ovládání a databázi pro ukládání dat. A to je v podstatě cílem této diplomové práce. Konkrétně je úkolem navrhnout a implementovat Front-end webového rozhraní společně s vytvořením databáze. Diplomová práce bude rozdělena do několika kapitol.

Cílem první kapitoly bude seznámení se s technologiemi potřebnými pro zhotovení této práce. Například technologiemi zabývajícími se vývojem moderních webových stránek jako Bootstrap nebo samotné HTML, které tvoří základní strukturu webových stránek. Mimo webové technologie se kapitola (2) bude zabývat i použitou platformou, na které by mělo zařízení měřící tlak fungovat. Konkrétně se bude jednat o oblíbenou platformu Raspberry Pi.

Mimo webové technologie bude diplomová práce v kapitole (3) řešit i seznámení se s principy měření tlaku a různé typy senzorů. A to důvodu lepšího přehledu nad celkovým měřícím řetězcem pro měření tlaku. Například bude řešit funkci kapacitního senzoru tlaku, který je v projektu použit.

Hlavní cíl praktické části diplomové práce, který bude rozebrán v kapitole (4), bude samotný návrh a implementace webové stránky společně s vytvořením databáze. A to od analýzy původního konceptu po použití webových technologií a celkovou implementaci jednotlivých komponent. Součástí je také i návrh jednotlivých tabulek databáze a následný popis ukládaných dat.

Poslední kapitola (5) se bude zabývat samotným testováním webové stránky a následným ošetřením nechtěných chyb, což vylepší celkovou kvalitu webové aplikace.

2 Seznámení se s použitými technologiemi

V této kapitole budou popsány technologie, které se pojí s praktickou částí této práce nebo mají s tímto tématem úzkou spojitost. Dále bude také znázorněno několik příkladů základního použití těchto technologií. Mezi základní technologie patří:

- **HTML** – poskytuje základní strukturu stránek, která je vylepšena a modifikována jinými technologiemi, jako je CSS a JavaScript,
- **CSS** – slouží ke kontrole prezentace, formátování a rozvržení,
- **JavaScript** – používá se k ovládání chování různých prvků.

2.1 HTML

HTML je neustále vyvíjející se hypertextový programovací jazyk, který byl vytvořen koncem osmdesátých let fyzikem Timem Berners-Lee a informatikem Robertem Caillau v CERNu¹, kde vznikl i první webový server. Základní myšlenka HTML byla sdílení libovolného textu, který mohl odkazovat na jiný libovolný text(hypertext). Později se podařilo Timu Berners-Lee tuto myšlenku prosadit do světového měřítka. První verze tohoto jazyku uměla jen několik příkazů na zvýraznění strukturovaného textu a hypertextové odkazy. Tento programovací jazyk se dále rozvíjel do následujících verzí.[1]

- **HTML 1** – První neoficiální specifikace jazyka a ovladač software uvolněné v roce 1991.
- **HTML 2** – Na této verzi se začalo pracovat již v roce 1993, kdy hlavní zaměření bylo sjednotit všechny značky a vytvořit ucelený standart, kterým by se řídili všichni. Mimo jiné přidává i prvky pro vytváření formulářů. Finální verze tohoto rozšíření byla hotová až v roce 1995.
- **HTML 3.0** – Na vývoji tohoto neoficiálního rozšíření pracovala firma Netscape už od roku 1994. Rozšíření bylo určeno pro prohlížeče Netscape Navigator. Dokončeno bylo v roce 1995.
- **HTML 3.2** – Tato verze je opět snahou sjednotit předchozí verze a vytvořit standart. Spousta rozšíření z neoficiální verze HTML 3.0 nebyla zahrnuta. Vývoj HTML 3.2 započal v roce 1996 a jeho oficiální verze byla vydána o rok později. Mezi hlavní novinky patří například možnost vytvářet tabulky.

- **HTML 4.0 a 4.01** - Na vývoji této verze se začalo pracovat souběžně s vydáním HTML 3.2. Tato generace přinesla hlavně nové výkonné funkce pro tvorbu rámců. Dále také podporu

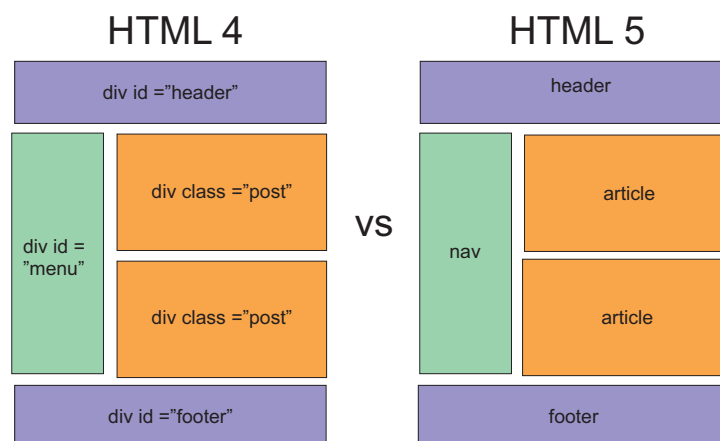
¹CERN (z franc. Conseil Européen pour la recherche nucléaire) Evropské výzkumné centrum částicové fyziky.

Javascriptu (2.3) nebo vkládání objektů jako například Flash² a Applet³. U této verze bylo zredukováno velké množství tagů, které neměly významovou hodnotu. HTML 4.0 se dále dělí na tři větve, tyto větve mají odlišná přísná pravidla. Tyto pravidla jsou:

- **Strict** zahrnuje elementy a atributy HTML s výjimkou elementů, které mají zobrazovací hodnotu. Nejsou povoleny rámce.
- **Transitional** oproti *strict* povolují elementy se zobrazovací hodnotou.
- **Frameset** povolují jak elementy se zobrazovací hodnotou, tak rámce.

Verze HTML 4.01 byla jen drobná aktualizace opravující chyby.[1]

• **XHTML** – je jedna z větví HTML 4, která původně měla rozšiřovat. X v názvu znamená eXtensible (rozšiřitelný), ve skutečnosti ale zužuje a omezuje. Je to forma XML, která splňuje normy XML jazyka. Nakonec byla tato větev opuštěna z důvodu přirozeného rozvoje webu a také kvůli absenci přínosu. [1]



Obrázek 1: Rozdíl mezi HTML 4 a HTML 5.[2]

• **HTML 5** – je v současnosti poslední vývojový stupeň jazyka HTML (poslední specifikace HTML 5.2 je z roku 2017). Mezi hlavní přínosy této verze patří nové sémantické prvky jako `<header>`, `<footer>`, `<article>` a `<section>`. Atributy prvků formuláře, jako je číslo, datum, čas, kalendář a rozsah. Grafické prvky `<svg>` a `<canvas>`. Multimediální prvky `<audio>` a `<video>` a mnoho dalšího. Od 5. verze HTML se již neočekává další vývoj, bude se však pokračovat dále ve vylepšování jednotlivých částí. [1]

²Flash je grafický vektorový program. Používá se především pro tvorbu (převážně internetových) interaktivních animací, prezentací a her.

³Applet je softwarová komponenta, která běží v kontextu jiného programu (webového prohlížeče nebo v panelu pro přepínání oken v grafickém uživatelském prostředí).

Použití HTML

HTML je jádrem jakéhokoliv webu bez ohledu na složitost a počet zapojených technologií. Patří mezi základní prvky každého profesionálního webu. Patří mezi takzvané značkovací jazyky (Markup languages). Tyto značky se používají k označení typu obsahu jako například odstavce, obrázku, odkazu nebo i nového řádku. Těmito značkami je pak tvořena každá webová stránka, kdy každý typ obsahu je „zabalěn“ neboli obklopen do takzvaných tagů. Například tagem `<p>` je označen odstavec.

Základní struktura každého webu začíná `<!DOCTYPE html>`. Tímto každý prohlížeč ihned pozná, že se jedná o strukturu HTML. Pomocí tagu `<body>` označujeme tělo webu, do kterého vkládáme veškerý obsah jako například odstavce, obrázky, odkazy, a tak dále. Tento obsah už musí být označován pomocí tagů tak, aby web dostal nějakou strukturu. Základní strukturu webové stránky můžeme vidět ve výpisu kódu 1, který je doplněn navíc o tagy `<h1>` (označující nadpis 1.úrovně) a již zmíněný `<p>` (označující odstavec).

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Nadpis</h1>
    <p>Odstavec</p>
  </body>
</html>
```

Výpis 1: Základní struktura webového dokumentu

2.2 CSS (Cascading Style Sheets)

Česky kaskádové styly. Tento programovací jazyk určuje vzhled, jakým se HTML prvky zobrazují na front-endu webové stránky. Prvotním autorem byl *Håkon Wium Lie*, poté byl jazyk navržen standardizační organizací W3C⁴. Bylo vydáno několik revizí, viz rozdělení níže. Cílem CSS bylo odstranění veškerého formátování z dokumentu HTML a následně přesunutí do dokumentu CSS (.css) z důvodu větší přehlednosti kódu. [1]

- **CSS 1** – Tato první verze CSS jazyku vznikla v roce 1996 a obsahovala již selektory a pravidla pro základní úpravu a pozicování textu.

⁴W3C – mezinárodní konsorcium, které společně s veřejností vyvíjejí standardy pro World Wide Web.

- **CSS 2.1** – U této verze došlo k rozšíření v oblasti pozicování textu a zamezení nutnosti měnit obsah kvůli formě. Dokončení této verze přišlo až v červnu roku 2011.

- **CSS 3** - Nejnovější verze, která se již pojí s nástupem HTML 5 popsaném v předchozí kapitole (2.1). Přidává další pokročilejší metody zobrazení jako například transformace 2D, ale také i 3D a animace. Mezi další rozšíření patří i řada nových atributů, selektorů a pseudoselektorů nazývané také *queries*, které dokážou rozlišit různé druhy zařízení a přizpůsobit podle toho styl webu. Stejně jako HTML 5 se tento jazyk neustále vyvíjí.[1]

2.2.1 Použití CSS

Jednoduše řečeno CSS je seznam pravidel, které mohou HTML tagům přiřadit různé vlastnosti, ať už se jedná o jednotlivé značky, více značek, celý dokument nebo více dokumentů. Propojit CSS s HTML můžeme třemi způsoby a to inline, interně a externě. Syntaxe těchto způsobů je uvedena na třech vzorových příkladech. První příklad, viz výpis(2), ukazuje inline způsob zápisu stylu, který v tomto případě nastavuje barvu celého odstavce na červenou. Druhý příklad, viz výpis(3), stanovuje na rozdíl od předchozího příkladu, kdy byl obarven pouze jeden odstavec, že všechny odstavce v HTML dokumentu budou obarveny modrou barvou s výjimkou, kdy bychom si jednotlivé odstavce nastavili podle sebe, jak je to u předchozího příkladu. Třetí příklad, viz výpis(4), je v podstatě stejný jako druhý s tím rozdílem, že jsou styly uloženy v jiném souboru, například „style.css“.

```
<p style="color: red">text</p>
```

Výpis 2: Příklad inline zápisu CSS kódu

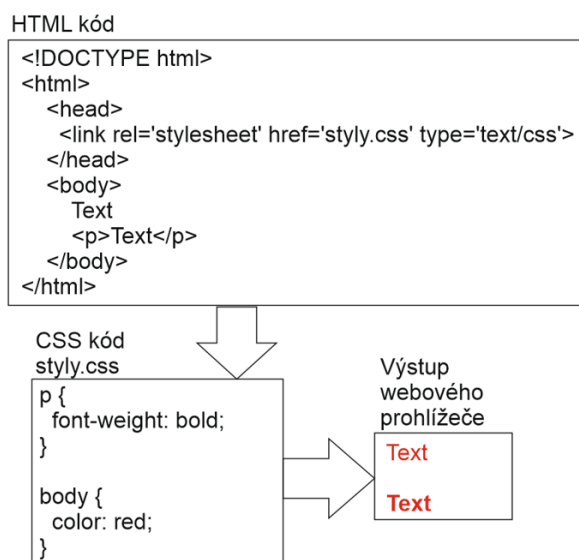
```
<head>
  <style>
    p {
      color: blue;
    }
  </style>
</head>
```

Výpis 3: Příklad interního zápisu CSS kódu

```
<head>
  <link rel='stylesheet' href='style.css' type='text/css'>
</head>
```

Výpis 4: Příklad externího zápisu CSS kódu

Na závěr je na obrázku (2) ukázka propojení jednotlivých logických vrstev stylu a HTML dokumentu. Na tomto příkladu můžeme vidět, jak CSS může měnit celkový vzhled dokumentu, kdy máme nastaveno, že všechny text v těle HTML bude červený a text, který je v ostavcích, je navíc ještě zvýrazněn.



Obrázek 2: Ukázka propojení logických urovní CSS a HTML.

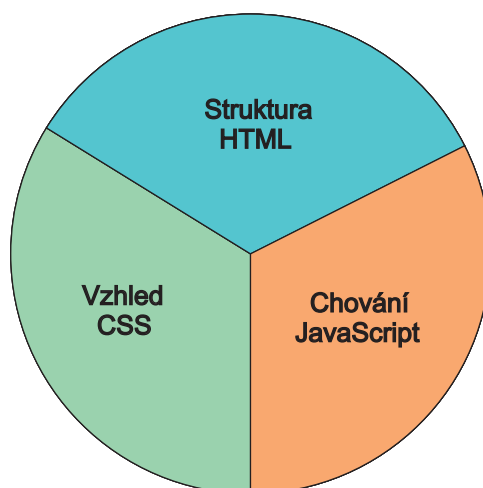
2.3 JavaScript

JavaScript je komplikovanější jazyk než HTML nebo CSS. Jeho autorem je *Brendan Eich*, který pracoval pro společnost Netscape. Netscape měla snahu vytvořit jazyk, který bude jednodušší, než třeba Java, a aby bylo tudíž jednodušší i použití pro webové vývojáře. Původní jméno tohoto jazyka bylo Mocha, které bylo změněno na JavaScript v roce 1995 a v roce 1997 Asociace ECMA vydala standardizovanou verzi JavaScriptu, kterou pojmenovala názvem ECMAScript, který je dodnes oficiálním názvem, i když se stále častěji používá původně obchodní název JavaScript.

V současné době je JavaScript podporován všemi moderními webovými prohlížeči a je používán téměř všemi webovými stránkami, které potřebují komplexnější funkce.

JavaScript je skriptovací a multiplatformní jazyk (to patří jeho hlavní výhody). S určitými omezeními umožňuje i objektově orientovaný přístup. Syntaxi tohoto jazyku řadíme do rodiny C/C++/Java, ale principiálně je sémanticky odlišný. Má rysy jak imperativního, strukturovaného, dynamického a funkcionálního jazyku, tak, jak již bylo řečeno, i objektově orientovaného jazyku.

Většina složitějších a moderních webových stránek je sestavena ze zmíněných webových technologií (HTML, CSS a JavaScript), viz obrázek (3), díky kterým mohou být webové stránky interaktivní, dynamické, nebo mohou fungovat i jako program. První ze základních stavebních kamenů HTML zabezpečuje základní strukturu celého webu nebo webové aplikace. Další CSS má na starost, jak již bylo zmíněno, celkový vzhled všech použitých částí. A jako poslední stavební kámen JavaScript vytváří z klasického webu webovou aplikaci (například vykreslení nějakých dat, které se obnovují v daném intervalu).[3]



Obrázek 3: Znázornění základních kamenů moderní webové stránky.

Imperativní a strukturovaný jazyk – Imperativní programování je speciálním paradigmatem strukturovaného programování, které popisuje průběh programu pomocí posloupnosti příkazů neboli algoritmus, jak řešit daný úkol. Strukturované programování přináší navíc větvení a cykly programu, které úkol programu rozloží do určitých funkcí s parametry. Tyto funkce jsou pak podle potřeby volány z hlavního programu. Kompilaci těchto technologií provádí samotný prohlížeč.[3]

Dynamický jazyk – Tento jazyk funguje tak, že v průběhu své činnosti provádí většinu věcí, které by statické programovací jazyky dělaly během doby kompilace.[3]

Funkcionální jazyk – Vyhodnocuje programovací paradigma jako matematický výpočet, který můžeme považovat jako funkci i argument. Tyto výsledky těchto argumentů můžeme dále přiřazovat do proměnných. [3]

2.3.1 Objektově orientované programování (OOP)

Jedná se o nový styl, filozofii, způsob myšlení, implementace a designu, kde je hlavní důraz kladen na znovupoužitelnost. Nový přístup je inspirován průmyslovou revolucí. Základem je vymyšlení

jednotlivých součástí celku, které budeme používat znovu a znovu (například, když stavíme židli, tak některé komponenty se opakují, jako šroubky nebo kolíky, takže je nebudeme muset vymýšlet znova a znova, ale pouze jednou). Stejně tak potom sestavujeme samotný program. Díky tomuto je přehlednější, rychlejší a levnější. Na komponenty můžeme v podstatě nahlížet jako na černou skříňku, kterou může naprogramovat někdo jiný, a my ji jen použijeme.[4]

• Implementace JavaScriptu do HTML

Stejně jako u CSS máme více způsobů implementace JavaScriptu, a to inline, interně a externě. Inline není doporučeno používat z důvodu nerespektování oddělení logických vrstev, navíc je značně neefektivní. U interní možnosti implementace, která se vyplatí spíše u menších webů nebo při učení této technologie, se program vloží mezi HTML tagy *script*. Externí je poslední a nejlepší způsob implementace JavaScript kódu. Propojení s HTML je stejně jako u interní implementace provedeno pomocí tagu *script* s tím rozdílem, že do argumentu *src* zadáme adresu JavaScriptového souboru(.js), viz příklad (5). Díky tomuto si můžeme logicky oddělit části webové aplikace a zabezpečit větší přehlednost kódu.

```
<head>
  <script src="script.js"></script>
</head>
```

Výpis 5: Příklad externího zápisu JavaScript kódu

2.4 PHP

PHP dnes známe pod produktovým označením PHP/FI. Bylo vytvořeno roku 1994 *Rasmus Lerdorfem*. První verze PHP byla jednoduchá sada Common Gateway Interface (CGI) napsaná v jazyce C. Původně byl použit pro sledování návštěvnosti online životopisu. Sada scriptů je pojmenována jako „Nástroje osobní domovské stránky“, častěji „nástroje PHP“. Časem však bylo požadováno více funkcí. *Rasmus* přepsal PHP nástroje a vytvořil mnohem větší a bohatší implementaci. Tento nový model byl schopný interakcí s databází a poskytoval tak uživatelům framework, na němž mohli jednoduše vyvíjet dynamické webové aplikace, jako například *guestbooks*⁵. V červnu roku 1995 vydal Rasmus PHP tools jako opensource, což vývojářům umožnilo používat tento nástroj, jak uznali za vhodné. To vedlo jak k opravě chyb, tak k vylepšení kódu a zároveň ke zdokonalení uživatelů.

V září téhož roku *Rasmus* rozšířil PHP a na krátkou dobu zrušil jméno PHP a nahradil jej jménem FI (zkratka pro „Forms Interpreter“). Tato nová implementace zahrnovala některé základní funkce z PHP, které známe dodnes. Měl podobné proměnné jako Perl, automatickou interpretaci proměnných formulářů a vestavěnou syntaxi jazyka HTML. I samotná syntaxe byla

⁵guestbooks – návštěvní knihy

hodně podobná Perlu, i když mnohem omezenější, jednodušší a poněkud nekonzistentní. Ve skutečnosti, aby kód vložili do souboru HTML, museli vývojáři použít komentáře HTML. Přestože tato metoda nebyla zcela dobře přijata, FI pokračovala v růstu a akceptaci jako nástroj CGI, ale přesto ne jako jazyk. Tohle se však změnilo v následujícím měsíci, kdy v říjnu roku 1995 Rasmus vydal kompletní předělávku kódu a také vrátil název na PHP, což ve stručnosti znamenalo „Personal Home Page Construction Kit“ a bylo to první vydání, které se mohlo chlubit pokročilým skriptovacím rozhraním. Návrh byl záměrně navržen tak, aby připomínal strukturu jazyka C z toho důvodu, aby bylo jednoduché si jej osvojit pro vývojáře, kteří již s jazykem C, Perl a podobnými jazyky pracovali. Jelikož byl prozatím omezen na systémy kompatibilní se systémy UNIX a POSIX, byl potenciál pro operační systém Windows NT předmětem implementace a zkoumání.

Kód byl znova kompletně předělán v dubnu roku 1996, kdy Rasmus představil nové jméno, které zkombinoval z předchozích vydání, a to PHP/FI. U této druhé generace skutečně došlo k tomu, že se PHP začalo vyvíjet ze sady nástrojů do samostatného programovacího jazyka. Obsahovala vestavěnou podporu databázi DBM, MySQL a Postgres95, cookies, uživatelsky definované funkce a mnoho dalšího.

V letech 1997 až 1998 byla kultura PHP rozšířena po celém světě, kde jej využívalo několik tisíc uživatelů. Podle průzkumu internetové společnosti Netcraft bylo zjištěno, že okolo šedesáti tisíc domén obsahovalo hlavičku PHP. V celkovém měřítku bylo přibližně 1% spuštěných PHP domén.[5]

• PHP 3

PHP 3.0 byla první verze, která se podobala dnešnímu PHP. Zjistilo se, že PHP/FI 2.0 stále postrádá funkce, které potřebuje internetová aplikace pro elektronické obchodování, která byla vyvíjena jako univerzitní projekt.

V době, kdy Rasmus chtěl přejít na online PHP, tak se diskutovalo o různých aspektech současné implementace a jejich rekonstrukci. Ve snaze zlepšit chod stávající základní PHP/FI se rozhodli Andi, Rasmus a Zeev vyvíjet nový nezávislý programovací jazyk.

Tento zcela nový jazyk byl vydán pod novým názvem, který odstranil důsledky omezeného osobního užití, které obsahovalo jméno PHP/FI 2.0. Nový název byl jednoduše PHP, což znamená, že se stalo rekurzivním akronymem – PHP: Hypertext Preprocessor.

Jednou z nejsilnějších možností PHP 3.0 bylo její silné rozšíření. Kromě toho, že poskytuje koncovým uživatelům zralé rozhraní pro více databází, protokolů a rozhraní API, usnadňují rozšíření samotného jazyka desítky vývojářů, kteří vyvíjeli různé moduly. Pravděpodobně to byl

klíč k úspěchu. Další klíčové funkce představené v PHP 3.0 zahrnovaly objektově orientovanou programovou podporu a mnohem výkonnější a konzistentnější jazykovou syntaxi.

V červnu 1998, kdy se připojilo k vývoji php mnoho nových programátorů z celého světa, oznámil vývojový tým PHP 3.0 jako oficiálního nástupce PHP 2.0, u kterého se vývoj ukončil předchozí rok. Po zhruba devíti měsících veřejného testování, kdy bylo oznámeno oficiální vydání PHP 3.0, bylo již nainstalováno na více než 70 000 doménách na celém světě a nebylo omezeno na operační systémy kompatibilní s POSIX. Poměrně malá část domén, hlásících PHP jako nainstalované, byly hostovány na serverech se systémem Windows 95, 98, Windows NT a Macintosh. Na svém vrcholu byl PHP 3.0 nainstalován na přibližně 10% webových serverů na internetu. [5]

• PHP 4

V zimě roku 1998, krátce potom kdy byl realizován PHP 3.0, Andi Gutmans a Zeev Siraski začali pracovat na přepsání PHP kódu. Cílem projektu bylo zlepšit výkonnost aplikací a zlepšit modularitu PHP základny. Tyto aplikace byly možné díky novým funkcím PHP 3.0 a podporou pro širokou škálu databází a API třetích stran, ale PHP 3.0 nebylo navrženo tak, aby efektivně zpracovávalo složité aplikace.

Nový engine nazvaný Zend Engine (složený z jejich prvních jmen, Zeev a Andi) úspěšně splnil své cíle a byl prvně představen v polovině roku 1999. PHP 4.0 založený na tomto enginu byl spolu s novou řadou funkcí oficiálně vydán v květnu 2000, téměř 2 roky po jeho předchůdci. Kromě lepšího výkonu PHP 4.0 zahrnoval další klíčové funkce, jako je podpora pro mnoho webových serverů, http relací, bufferování výstupu, bezpečnější způsoby manipulace s uživatelskými vstupy a několik nových jazykových konstrukcí. [5]

• PHP 5

PHP 5 byl spuštěn v červenci roku 2004 po dlouhém vývoji a několika předběžných verzích. Je poháněn jádrem Zend Engine 2.0 s novým objektovým modelem a desítkami dalších nových vlastností.

Vývojový tým společnosti PHP zahrnuje mnoho vývojářů, kteří pracují na projektech týkajících se PHP a jiných podpůrných projektech, jako jsou PEAR, PECL a dokumentace, a základní síťovou infrastrukturu více než sta jednotlivých webových serverů na šesti ze sedmi kontinentů světa. Přestože je odhad založený pouze na statistikách z předchozích let, lze předpokládat, že PHP je nyní instalováno na desítkách nebo dokonce i na stovkách milionů domén po celém světě. [5]

• **Ukáža PHP kódu** – konkrétně je na příkladu (6) znázorněn způsob propojení webové aplikace s SQL databází.

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Vytvoreni pripojeni
$conn = new mysqli($servername, $username, $password);

// Kontrola pripojeni
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Výpis 6: Ukázka PHP kódu

2.5 Bootstrap

Styl a prezentace webové stránky hraje důležitou roli při tvorbě webových stránek. Z tohoto důvodu se neustále vyvíjejí prostředky pro získání moderního a atraktivního vzhledu webových stránek. Kromě vzhledu je kladen důraz i na rychlost a jednoduchost použití. Přesně takové požadavky daly vzniku frameworku Bootstrap.[8]



Obrázek 4: Logo Bootstrapu.[9]

Bootstrap vytvořili zaměstnanci Twitteru v roce 2010, kvůli snaze sjednotit nekonzistentní aplikace v jejich firmě a ke snížení potřeby znalosti několika různých CSS stylů. Z toho důvodu začali pracovat na CSS frameworku jménem Twitter Blueprint, který později firma uvolnila jako open source. Bootstrap je jedním z nejpopulárnějších front-end frameworků a open source

projektů na světě. Tento framework v podstatě kopíruje nejmodernější trendy v designu webu.[6]

Použití frameworku je pokládáno za stejný skok ve vývoji tvoření webových stránek, jako je používání oddělených logických struktur webu jako například CSS nebo JavaScript. Framework Bootstrap je v podstatě CSS a JavaScriptová knihovna využívající ke své funkci také jQuery. Bootstrap nabízí ty nejmodernější trendy tvorby webových stránek, mezi které patří zejména:

- **Znovupoužitelnost** – Při navrhování webové stránky je modulární vzor upřednostňován, protože není potřeba měnit kód pro různé části návrhu. Bootstrap již obsahuje velkou část komponent a stylů připravených pro použití. Tento aspekt ušetří značné množství času a úsilí, což vede k rychlému vývoji webových stránek, ke zlepšení přehlednosti kódu a tím i ke snadnější údržbě kódu a organizaci struktury.[8]

- **Konzistence** – Snadná čitelnost kódu je pro návrháře velice důležitá. Pomáhá také k tomu, aby mohlo více návrhářů pracovat na stejném projektu a dobře chápali kódy jiných lidí, se kterými spolupracují. A to tak, že jsou schopni provádět modifikace a úpravy kódu. Vzhledem k tomu, že Bootstrap používá hotové fragmenty kódu a je kompatibilní s různými prohlížeči, je v procesu vývoje vysoká úroveň identičnosti. To vede ke zkrácení doby potřebné k zaučení nových vývojářů a implementaci podobné struktury na rozdílných projektech.[8]

- **Flexibilní rozvržení mřížky** – Bootstrap má výchozí mřížkový systém, který můžeme škálovat až do dvanácti sloupců s relativním zvětšením velikosti v závislosti na velikosti zobrazovacího monitoru. Kromě toho umožňuje Bootstrap přidání libovolného počtu vlastních sloupců. Tyto sloupce v závislosti na velikosti monitoru pomocí vestavěné funkce mixins naskládá buď za sebe nebo pod sebou.[8]

- **Přizpůsobení** – Bootstrap umožňuje také možnost vlastního přizpůsobení pomocí vestavěné volby `Customize`, kde je možné si vybrat, které funkce chceme použít. Je také možné použít vlastní šablony CSS k přepsání výchozích stylů. Kromě stylů si můžeme přizpůsobit i chování pluginů jako Modals a Alerts a to pomocí pokročilého JavaScriptu. V nejnovější verzi Bootstrapu je možné také vytvářet komplexní a interaktivní webové stránky.[8]

- **Komunita** – Bootstrap má k dispozici velmi rozsáhlou podporu třetích stran, kde dochází k neustálé improvizaci a vývoji nástrojů pro správné používání Bootstrapu. Mezi takové programy patří například `Bootlint`, který dokáže vyhledat v kódu nesprávné použití Bootstrapu. Dalším nástrojem vyvinutým komunitou je `Node`, který slouží jako správce balíčků pro Bootstrap.[8]

Tabulka

ABC
1 4 1
5 2 1
1 6 3
[Zobrazit více informací](#)

Přihlaste se

Zadejte e-mail:



Tabulka

A	B	C
1	4	1
5	2	1
1	6	3

[Zobrazit více informací](#)

Přihlaste se

Zadejte e-mail:

Obrázek 5: Ukázka aplikování frameworku Bootstrap.

- **Futuristický výhled a otevřený vývoj** – Vývoj Bootstrapu je explicitně prováděn na GitHubu, kde je možné sledovat veškeré provedené aktualizace a sledovat záznamy o nevyřešených problémech souvisejících s vývojem Bootstrapu. [8]

Na obrázku (5) můžeme vidět využití Bootstrapu. V horní části obrázku můžeme vidět webovou stránku bez stylů obsahující tabulku, hypertextový odkaz, textové pole a tlačítko. Jakmile byl vložen framework Bootstrap, můžeme na spodní části obrázku vidět, jak stránka dostala reprezentativnější vzhled a také schopnost přizpůsobovat se velikosti zobrazovací plochy.

2.6 jQuery

jQuery je velice populární javascriptovou knihovnou (sadou funkcí), která především ulehčuje vývoj v jazyce javascript. Hlavním cílem této knihovny je sjednotit implementaci javascriptu pro všechny platformy. To znamená, že kód bude fungovat na všech webových prohlížečích stejně. Dalším cílem je jednoduchost, rychlost a čitelnost kódu. jQuery je dostupný zdarma dokonce s velmi přehledně popsanou dokumentací a návody. [10]

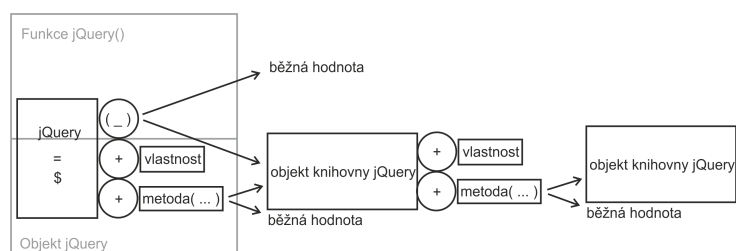
jQuery byl vydán v roce 2006 autorem Johnem Resigem. Důvodem byla především touha ulehčit si práci a to v tom směru, že Johna Resiga nebavilo bojovat s rozdíly mezi webovými

prohlížeči. Mimo jiné také touto knihovnou vylepšil práci s hledáním a manipulací s elementy dokumentu HTML.[11]

Základním stavebním kamenem knihovny jQuery je stejnojmenná funkce jQuery(). Tato funkce se chová jako každá jiná, má své parametry a nějakou návratovou hodnotu. Funkce má mnoho variant, které se liší právě počtem nebo typem parametru a návratové hodnoty. Funkce jQuery() má v programu svůj alias \$(). Mimo jiné můžeme mít jQuery() jako objekt, který je ve skutečnosti také funkce, jen z jiného pohledu. Znázornění jednotlivých pojmů můžeme vidět na obrázku (6). [11]

Jako poslední informaci k jQuery můžeme uvést pár základních dovedností této knihovny, ty jsou:

- vybrat a změnit objekty DOM,
- události,
- manipulace s CSS,
- selektory,
- efekty (pomocí předdefinovaných funkcí),
- animace - jednoduchá tvorba a velmi efektní i efektivní výsledek,
- ajax - načítání obsahu serveru bez nutnosti obnovení stránky,
- spousta pluginů,
- utility – např. informace o prohlížeči nebo funkce each,
- spousta dalších věcí. [11].



Obrázek 6: Znázornění pojmů jQuery.

• **Ukázka použití jQuery** – Nejčastěji používaný kód v jQuery můžeme vidět na výpisu kódu (7). Tento kód volá funkci jQuery, která říká, že po dokončení načtení stránky je vykonán kód uvnitř této funkce.

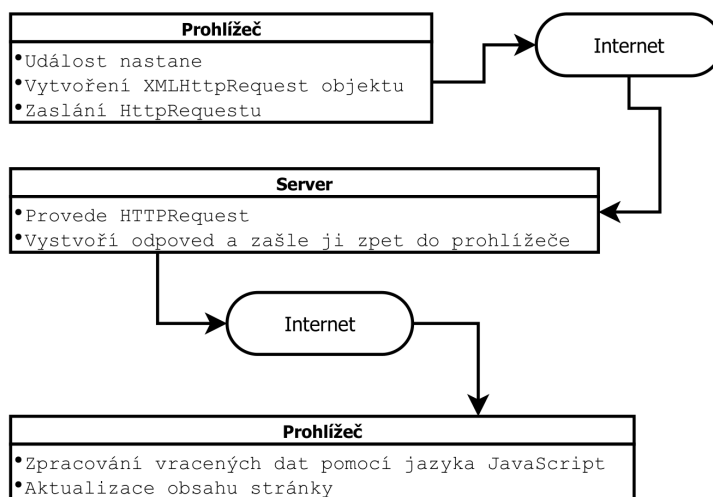
```
<script type="text/javascript">
  $(document).ready(function(){
    //Udelej neco
  });
</script>
```

Výpis 7: Ukázka jQuery kódu

2.7 AJAX

AJAX je zkratka pro asynchronní JavaScript a XML a popisuje soubor vývojových technik používaných pro vytváření webových stránek a webových aplikací. Podle webového vývojáře a instruktora Skillcrush WordPress Ann Cascarano je nejlepším způsobem, jak porozumět AJAXu, začít s identifikací jeho specifického účelu v procesu vývoje webu. Hlavním úkolem AJAXu je aktualizovat webový obsah asynchronně ("A" AJAXu), což znamená, že webový prohlížeč uživatele nemusí znovu načíst celou webovou stránku, pokud se potřebuje změnit pouze malá část obsahu na stránce.

Jako příklad použití asynchronní aktualizace si můžeme uvést vyhledávač Google **Google Suggest**. V případě zadání vyhledávacího dotazu se automaticky objeví nabídka možnosti automatického dokončování slov textu nebo klíčů vyhledávání. Funkce, jako je Google Suggest, jsou základní součástí současného prohlížení webových stránek, což ukazuje na to, jak důležitý je AJAX ve vývoji webových stránek. [12]

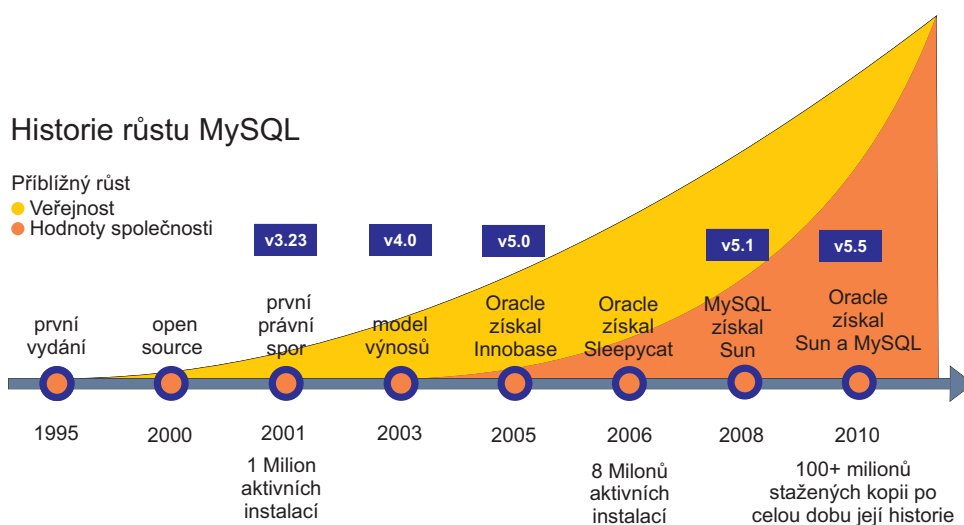


Obrázek 7: Příklad, jak funguje Ajax.

2.8 MySQL

MySQL je jedním z nejrozšířenějších open source systémů pro správu relačních databází na světě. Celková distribuce, která činí více než 100 milionů uživatelů po celém světě, se stala první volbou pro velké společnosti pro správu dat, které pokrývají širokou škálu internetových technologií. [13]

MySQL byl vytvořen švédskou společností MySQL AB v roce 1995. Vývojáři platformy byli Michael Widenius (Monty), David Axmark a Allan Larsson. Nejdůležitějším úkolem bylo poskytnout efektivní a spolehlivé možnosti správy dat domácím i profesionálním uživatelům. Více než půl tuctu alfa a beta verzí platformy bylo vydáno v roce 2000. Tyto verze byly kompatibilní s téměř všemi hlavními platformami. [13]



Obrázek 8: Vývoj MySQL.

Původně ve vlastnictví společnosti MySQL AB se stala v roce 2000 volně šiřitelná platforma, která dodržovala GPL⁶ podmínky. Přejít na open source měl za následek propad výnosu, který se však po čase vrátil zpět. Open source MySQL přirozeně otevřel cestu vývojářům třetí strany. [13]

MySQL získala postupem času čím dál větší popularitu jak u domácích uživatelů, tak i profesionálních. Již v roce 2001 měla platforma dva miliony aktivních instalací. V roce 2002 společnost rozšířila svůj dosah a otevřela americkou centrálu. Stejný rok bylo oznámeno, že počet uživatelů na této platformě přesahuje tři miliony. Tržby v tomto období se vyšplhaly až na 6,5 milionů dolarů. [13]

⁶GPL (General Public License) licence pro volně použitelný software

2.9 Raspberry Pi

Raspberry Pi je malý počítač velikosti kreditní karty navržený a vyráběný neziskovou organizací Raspberry Pi Foundation. Tato organizace se snaží zpřístupnit práci s počítačem a programování co možná největšímu počtu lidí. Ačkoliv prvotní cíl tohoto projektu bylo získání levného počítače s možnostmi programování pro studenty, byl tento počítač přijat mnoha elektrotechniky a programátory po celém světě. Pomocí Raspberry Pi se rázem začaly vytvářet miliony projektů, od herních arkádových automatů, různých robotů, až po použití jako domácí multimediální centru a mnoho dalšího.

První Raspberry Pi byl představen v roce 2012 (autorem Ebenem Uptonem, který se inspiroval mikropočítači BBC Micro⁷) pod označením **Raspberry Pi 1 Model A**. Představoval systém postavený na procesoru **Broadcom BCM2835**. Tento procesor byl malý a poměrně silný mobilní procesor používaný především v mobilních telefonech. Součástí procesoru byla GPU, zpracování audio/video a další funkce součástí čipu s nízkým odběrem běžícím na jednojádrovém procesoru ARM s kmitočtem 700 MHz. Během uplynulých let však nadace vydala více verzí, kdy Raspberry Pi osadila až čtyřjádrovým procesorem o kmitočtu 1.4 GHz.

Ačkoliv je Raspberry Pi úžasné a malé zařízení, které se od svého vzniku vyvíjelo mílovými kroky, je důležité zdůraznit, co Raspberry Pi není. Raspberry Pi není úplnou náhradou za stolní počítač nebo notebook. Například systém Windows není možné na Raspberry Pi spustit (alespoň ne v klasické verzi, kterou známe), přestože je možné spustit mnoho Linuxových distribucí s prostředím pro stolní počítače, webovými prohlížeči a dalšími prvky, které bychom očekávali od stolního počítače. Přestože Raspberry Pi úplně nenahradí veškeré výhody stolních počítačů nebo notebooků, je stále neuvěřitelně univerzálním zařízením se spoustou hardwaru za nízkou cenu ideálním pro hobby, elektroniku, učení programování a spoustu dalších experimentů.[14]

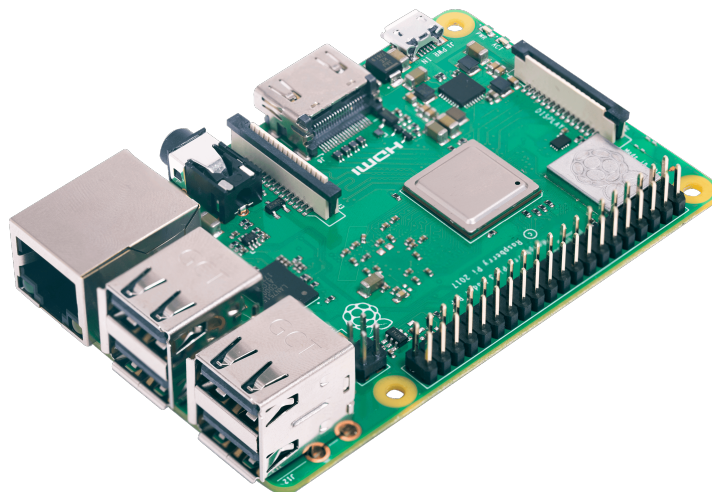
V prvních letech založení nadace bylo možné získat Raspberry Pi ve dvou verzích, a to Model A za 25 dolarů a Model B za 35 dolarů. Levnější varianta měla oproti dražší menší hardwarovou výbavu – o jeden USB port méně, absenci ethernetového portu a poloviční paměť RAM.

Vzhledem k tomu, že výrobní náklady klesaly a společnost Raspberry Pi Foundation získávala stále větší popularitu, dokázala výrazně zvýšit hardwarové specifikace zařízení a zároveň i udržet stejné náklady na výrobu. To vedlo k tomu, že veškeré další verze, ať už Raspberry Pi 2 z roku 2015 nebo Raspberry Pi 3 z roku 2016, se vždy prodávaly za maximální cenu 35 dolarů.

Současná generace Raspberry Pi 3 B+ disponuje hardwarem s těmito parametry:

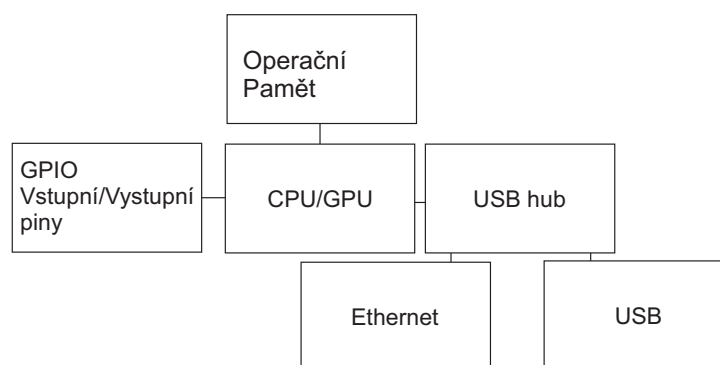
- 1.4 Ghz ARM processor Systems-On-a-Chip (SoC) s integrovanou 1 GB RAM,

⁷BBC Micro – Britská korporace vyrábějící sérii mikropočítačů a sdružených periférií v 80. letech pro zvýšení počítačové gramotnosti.



Obrázek 9: Raspberry Pi 3 B+.[16]

- HDMI port pro digitalní audio/video výstup,
- 3.5 mm jack, který nabízí jak audio, tak kompozitní video výstup,
- čtyřmi USB 2.0 porty pro připojení vstupních/výstupních periférii,
- čtečka karet microSD pro načtení operačního systému,
- Ethernet LAN port,
- Integrovaná bezdrátová anténa Wi-Fi / Bluetooth,
- Napájecí microUSB port,
- Rozhraní GPIO (General Purpose Input/Output).

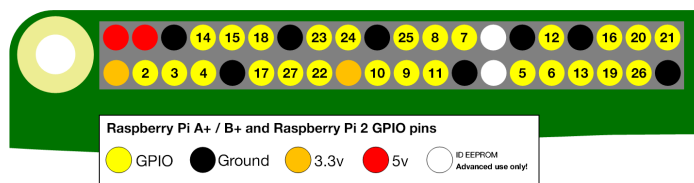


Obrázek 10: Blokové schéma Raspberry Pi.

GPIO

GPIO (General Purpose Input/Output) je jedním ze silných rysů tohoto malého počítače. Jedná se o řadu GPIO(univerzálních vstupních / výstupních) pinů podél horního okraje desky. Kterýkoli z pinů GPIO může být určen (v softwaru) jako vstupní nebo výstupní pin a použit pro širokou škálu účelů. GPIO rozhraní na desce Raspberry Pi obsahuje dva 5 V piny, dva 3,3 V piny, sérii uzemňovacích pinů. Zbytek pinů jsou programovatelné vstupy a výstupy. Programovatelné piny mohou nabývat hodnot 0 až 3,3 V a jejich tolerance je taktéž do 3,3 V. Dále je možné softwarově konfigurovat pull-up a pull-down rezistory, až na piny GPIO2 a GPIO3, které mají pevně dané pull-up rezistory. Kromě jednoduchých vstupních a výstupních zařízení mohou být piny GPIO použity s řadou alternativních funkcí, jako například:

- PWM (pulse-width modulation) – pulzně šířková modulace,
- SPI (Serial Peripheral Interface) – sériově periferní rozhraní,
- I2C (Inter-Integrated Circuit) – nízkorychlostní komunikace,
- Serial – Sériová komunikace.[15]



Obrázek 11: Rozhraní GPIO na Raspberry Pi.[15]

3 Seznámení se s měřením tlaku

Měření a řízení tlaku je v mnoha segmentech průmyslu nejpoužívanější procesní veličinou. Pomocí tlaku je možné odvodit řadu dalších parametrů, jako je například hladina, objem, průtok a hustota. Tato kapitola se bude zabývat hlavními technologiemi používaných při měření tlaku.

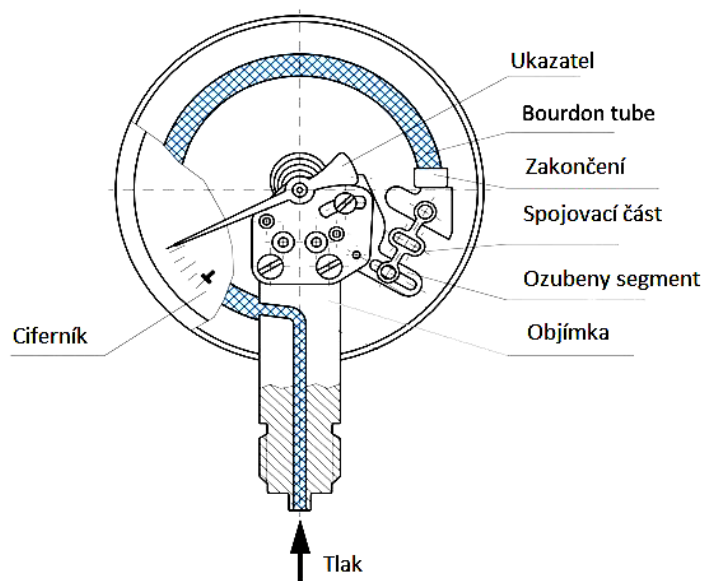
Měření tlaku po mnoho let vzbuzovalo zájem vědy. Na konci 16. století byl udělen patent italskému vědci Galileo Galilei za systém vodních čerpadel pro zavlažování. Pro zajímavost v roce 1592 Galileo vynalezl první teploměr fungující na principu podtlaku ve skleněné zkumavce, která byla naplněna do půlky vodou a otočená vzhůru nohama a napůl ponořena do vody. Takže při změně teploty vzduchu vzduch uvnitř trubky také změnil svoji teplotu a tím změnil i svůj objem. Jádrem zmíněného čerpadla bylo sací zařízení, které dokázalo vysát vodu do výšky maximálně deseti metrů. Galileo však nikdy nepoznal důvod tohoto limitu, ale jev byl natolik zajímavý, že motivoval ostatní vědce k tomu, aby jej studovali.

V letech 1643 italský fyzik Evangelista Torricelli vynalezl barometr. Díky tomuto objevu bylo možné zjistit atmosférický tlak, to znamená sílu vzduchu působící na zemský povrch. Jeho experiment spočíval v naplnění jednometrové z jedné strany uzavřené trubičky rtutí, která byla otevřeným koncem ponořena do vany s rtutí. Při tomto experimentu byla trubička držena svisle a na uzavřeném konci vždy poklesla hladina o 760 mm. Bez toho, aby přesně věděl, jaký je důvod tohoto chování, dospěl Torricelli k závěru, že prostor, který zanechala rtuť na uzavřené straně trubice, je prázdný a pojmenoval ho **vakum**.

O pět let později francouzský fyzik Blaise Pascal použil barometr, aby ukázal, že tlak se snižuje ze zvětšující se nadmořskou výškou. V roce 1650 německý fyzik Otto Von Guericke vyvinul první účinné vzduchové čerpadlo, které bylo Robertem Bourdonem využito na testování komprese a dekomprese. 200 let nato francouzský fyzik a chemik Joseph Louis Gay-Lussac definoval, že je-li zachována hmotnost a tlak plynu, pak jeho objem v závislosti na zvyšující se teplotě lineárně roste.

V roce 1849 získal Eugène Bourdon patent na tzv. Bourdon Tube (mechanické měřicí zařízení pro zjišťování tlaku plynu nebo kapaliny). [17]

Tlak patří mezi základní fyzikální veličinu používanou v technické praxi. Je označována písmenem p z latinského *pressura*. Hlavní jednotkou tlaku je Pascal **Pa**. Tato jednotka je velmi malá a v praxi se častěji používají násobky, jako například **hPa**, **kPa**, **MPa**. Stanovuje vlastnosti pevných nebo kapalných látek. Je několik způsobů vyhodnocení měřeného tlaku. Mezi nejčastější metody patří měření rozdílu měřeného tlaku k absolutnímu nulovému tlaku nebo k barometrickému tlaku. Absolutní nulový tlak je v podstatě absolutní vakuum a blíží se k nulové hodnotě.



Obrázek 12: Bourdonová trubice.[19]

V ideálním případě je prostor zbaven všech hmotných částic. Barometrický tlak se označuje p_b (někdy i p_0) a je to tlak, který je v atmosféře. Jeho hodnota je závislá jak na počasí, tak na nadmořské výšce. Čím výše se nacházíme, tím nižší je atmosférický tlak. Důležitým pojmem je také rozdílový tlak neboli tlaková diference. Jedná se o rozdíl dvou různých tlaků, přičemž se žádný z nich neshoduje s tlakem barometrickým. Při rozdílu měřeného tlaku vůči barometrickému tlaku dostáváme přetlak, který značíme p_e a je definovaný (1). [18]

$$p_e = p - p_0 \quad (1)$$

kde: p_e výsledný tlak,
 p absolutní naměřený tlak,
 p_0 barometrický tlak.

Hodnota přetlaku může nabývat kladných i záporných hodnot. Záleží, zda je měřený absolutní tlak vyšší nebo nižší, než hodnota tlaku barometrického. Podle toho, zda tlak nabývá kladných nebo záporných hodnot jej rozdělujeme na přetlak(kladný) a podtlak(záporný). Hrubé rozdělení tlaku můžeme vidět na obrázku (13).

3.1 Základní definice tlaku a jednotky

Tlak je v podstatě poměr síly dF , která působí ve směru normály na plochu dS . Výsledný tlak je tedy dán vztahem (2).

$$p = \frac{dF}{dS} \quad (2)$$

kde:	p	výsledný tlak,
	dF	síla,
	dS	plocha.

Tento vztah platí jak pro tlak kapaliny, tak i tlak plynů. U pevných těles z důvodu, že nemají ideálně hladkou styčnou plochu, představuje tlak nespecifikovanou průměrnou hodnotu. Z tohoto důvodu tento tlak nazýváme specifický tlak.

Jednotka tlaku, jak již bylo zmíněno, je jeden Pascal Pa. Dle definice platí, že 1 Pa je roven síle o velikosti 1 N, která působí rovnoměrně na plochu o obsahu 1 m^2 , viz (3).

$$1\text{Pa} = 1\text{N} \cdot \text{m}^{-2} \quad (3)$$

Kromě již zmíněných tlaků může například v kapalinách působit tlak statický a dynamický. Pokud je kapalina v klidovém stavu, tak v ní působí pouze hydrostatický tlak označený p_s . Tento tlak má stejnou sílu ve všech směrech a velikost této síly je úměrná výšce vodního sloupce. Hydrostatický tlak můžeme vyjádřit vztahem (4). [18]

$$p_s = \rho \cdot g \cdot h \quad (4)$$

kde:	p_s	hydrostatický tlak,
	ρ	hustota tekutiny,
	g	gravitační zrychlení,
	h	výška vodního sloupce tekutiny.

Pokud je ale kapalina v pohybu, tak kromě zmíněného hydrostatického tlaku musíme připočítat i tlak kinetický, označen p_k , tak i tlak dynamický p_d .

Kinetický tlak je daný rychlostí proudění a výsledek je tlakový účinek proudící tekutiny s danou hustotou.

$$p_k = \frac{w^2 \cdot \rho}{2} \quad (5)$$

kde:	p_k	kinetický tlak,
	w	rychlost proudění tekutiny,
	ρ	gravitační zrychlení.

U tekutin, které je možné stlačit, se kinetický tlak mění na dynamický dle následujícího vztahu(6).

$$p_d = p_k \cdot s \quad (6)$$

kde: p_d dynamický tlak,
 p_k kinetický tlak,
 s stlačitelnost kapaliny.

V případě nestlačitelných tekutin je stlačitelnost rovna jedné, tím pádem se shodují dynamické a kinetické tlaky.

$$p_d = p_k \quad (7)$$

Pokud předchozí dva tlaky (dynamický a kinetický) sečteme, dostaneme výsledný tlak proudu, který se nazývá celkový tlak p_c .

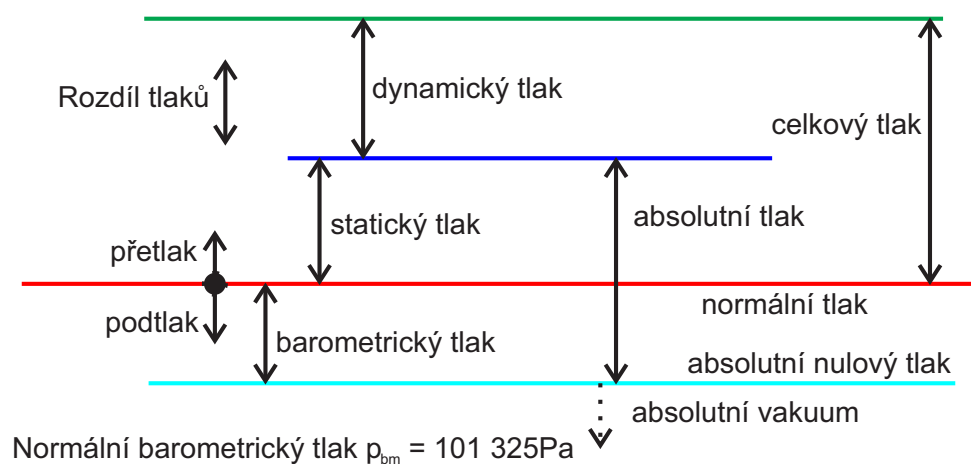
$$p_c = p_k + p_d \quad (8)$$

Při změně tlaku může docházet u kapalin také ke změně skupenství, tudíž je tlak také stavovou veličinou vyjádřenou ve vzorci (9).

$$p \cdot V = n \cdot R \cdot T \quad (9)$$

kde: p tlak,
 V objem,
 n látkové množství v molech,
 R univerzální tlaková konstanta,
 T termodynamická teplota.

Souhrn základních tlaků je znázorněn na obrázku 13.



Obrázek 13: Měření tlaků.

3.2 Senzory používané při měření tlaku

Obecně jsou senzory rozdělovány podle techniky měření tlaku. Všechny techniky mají jediný účel, a to převést sílu tlaku na elektricky úměrný signál. Mezi typy takových měřičů tlaku patří:

- tenzometrické (piezo odporové),
- piezoelektrické,
- rezonanční,
- kapacitní,
- potenciometrické,
- magnetické,
- optické.

Tenzometrické snímače tlaku

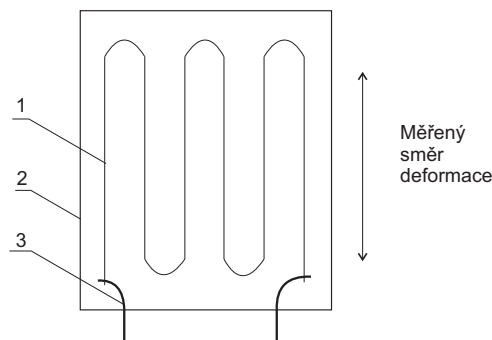
Princip tenzometrických tlakoměrů spočívá ve změně elektrického odporu vlivem deformace prvku v závislosti na síle tlaku. Intenzitu deformace převádějí ve většině případů na napěťový signál. Většinou se používá několik tenzometrů současně kvůli kompenzaci teploty okolí. Tenzometry můžeme rozdělit podle provedení na kovové a polovodičové. V průběhu deformace je měřen elektrický odpor, který je dán podle vztahu (10) odvozeného pro válcový vodič. Nevýhodou tenzometru je omezený rozsah provozních teplot – použitelnost pouze pro rozsahy nízkého tlaku. Příklad odporového tenzometru je znázorněn na obrázku (14). [18] [17]

$$R = \rho \cdot \frac{l}{S} \quad (10)$$

kde:	R	výsledný odpor vodiče,
	ρ	měrný odpor materiálu vodiče,
	l	délka válcového vodiče,
	S	plošný průřez vodiče.

Piezoelektrické snímače tlaku

Piezoelektrické snímače tlaku fungují tak, že přivedený tlak působí tlak na oddělovací membránu a ta deformuje piezoelektrický krystal. Tento krystal můžeme znát například ze zapalovačů a nejčastěji je vytvořen z křemíku. Princip spočívá v tom, že při deformaci tohoto krystalu vzniká změna elektrické polarizace. Díky tomuto jevu vzniká na krystalu náboj úměrný síle deformace působící kolmo na plochu zmíněného krystalu.



Obrázek 14: Znázorňuje drátový tenzometr, kde 1 je odporový drát, 2 je deformační podklad, 3 jsou konektory tenzometru. [18]

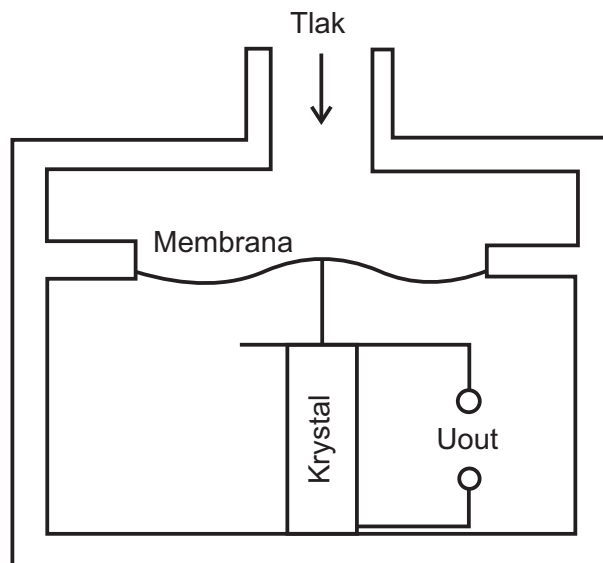
Jejich nevýhodou je, že vyžadují zapojení s vysokou impedancí a zesilovač s velkým zesílením. Dále jsou také citlivé na hluk a vibrace. Na druhou stranu mají výhodu v tom, že mají rychlé reakce na změny tlaku. Piezoelektrické tlakoměry jsou schopny měřit tlak až do velikosti 100 MPa.

Vztah pro získání velikosti náboje je popsán jako síla F působící kolmo na plochu monokrystalické destičky. Ta vlivem tlaku mění polarizaci P , viz vzorec (11). Dále můžeme odvodit napětí U vygenerované v závislosti na tlaku dle vzorce (12).

$$Q = P \cdot S = k \cdot F \quad (11)$$

$$U = \frac{Q}{C} = \frac{k \cdot a \cdot F}{\varepsilon_0 \cdot \varepsilon_r \cdot S} = k_u \cdot F \quad (12)$$

kde:	Q	elektrický náboj,
	S	plocha kolmá na směr působící síly,
	U	napětí vzniklé na monokrystalu,
	C	kapacita monokrystalické destičky,
	a	tloušťka destičky,
	k	piezoelektrická konstanta,
	ε_0	permitivita vakua,
	ε_r	permitivita použitého piezoelektrického materiálu,
	k_u	napěťová konstanta snímače.



Obrázek 15: Příklad znázorňující princip funkce piezoelektrického snímače tlaku.[21]

Rezonanční snímače tlaku

Princip funkce rezonančního tlakoměru vychází z rozkmitání membrány připojené k drátové cívce. Vlivem kmitání membrány vzniká magnetické pole a tím i elektrický proud. Frekvence kmitání je úměrná druhé odmocnině napětí na výstupu. Senzor bývá tvořen silikonovou kapslí upevněnou na membráně, která vibruje při rozdílu tlaků. Frekvence těchto vibrací je závislá na aplikovaném tlaku. [17]

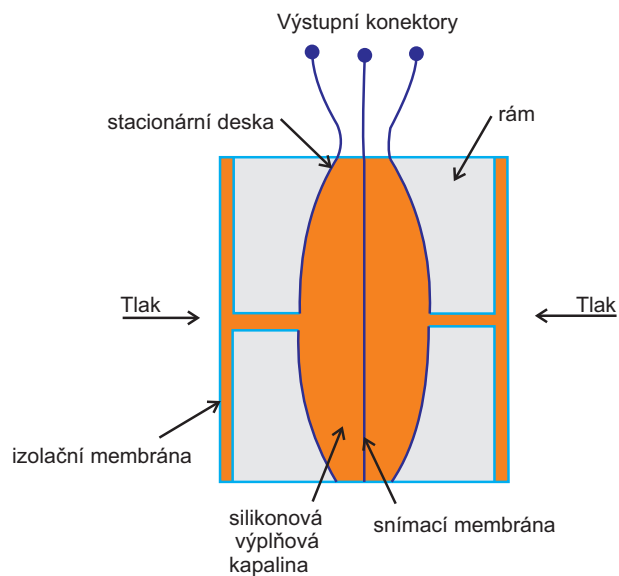
Kapacitní snímače tlaku

Kapacitní senzory tlaku jsou zpravidla jedny s nejspolehlivějších tlakoměrů a používají se na milionech zařízeních. U tohoto druhu snímače tlaku je deformačním prvkem membrána ve tvaru kruhu, která je zároveň taky uzemněnou elektrodou senzoru. Tato membrána je pevně uchycena v pouzdře tlakoměru. Druhá membrána je uchycena izolovaně a rovnoběžně s první. Při změně tlaku se kovová membrána pohybuje a tím se mění jak velikost vzduchové mezery, tak i kapacita.

Nevýhodou kapacity je závislost na teplotních dilatacích a nelinearita výstupu senzoru v závislosti na tlaku. Z toho důvodu se kapacitní snímač navrhuje jako diferenční. Tím jsou kompenzovány vlivy, které působí na obě části stejně. Jsou ideální jak pro nízkotlaké, tak i vysokotlaké použití.

Teplotní rozsahy, při kterých kapacitní senzory tlaku fungují, jsou určeny v závislosti na použité kapalinové výplni, a to:

- silikonový olej pro teploty -40 až 149 °C,
- upravený silikonový olej pro teploty 15 až 315 °C,



Obrázek 16: Diferenciální kapacitní senzor s oddělovací kapalinou. [20]

- fluorovaný olej pro teploty -45 až $205\text{ }^{\circ}\text{C}$.

Z obrázku (16) můžeme vidět, že diferenciální kapacitní senzor obsahuje snímací membránu. Membrána je tvořena velmi přesným pružinovým materiálem. Díky tomuto přesnému pružinovému materiálu můžeme velmi přesně předpovídat intenzitu působící síly na membránu. Na tuto membránu působí z obou stran dvě síly pomocí kapalinové výplně. Výsledný tlak je proto diferenční. [20]

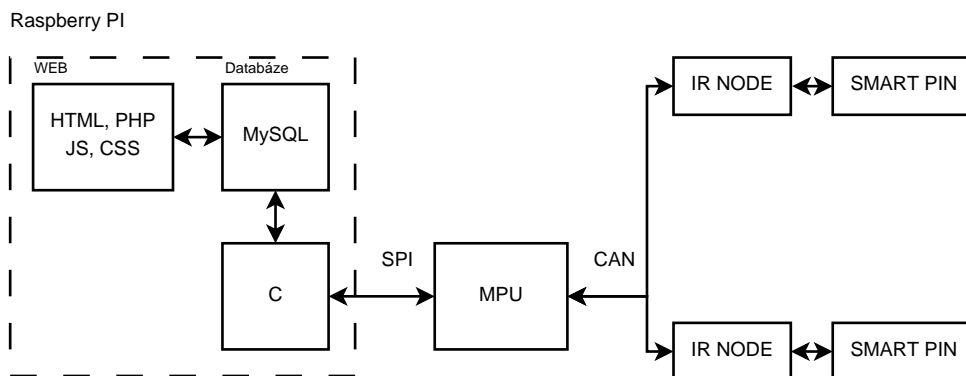
4 Návrh a implementace technického řešení front-endu

V této kapitole budou popsány detaily návrhu a implementace jak webového grafického rozhraní, tak databáze a v poslední řadě i obslužného programu.

Jako první úkol praktické části této diplomové práce byla volba technologie, která by byla schopna vizualizovat naměřená data, parametry a zároveň i tyto parametry měnit. Z počátku se rozhodovalo mezi standardními programovacími jazyky jako C++ nebo C# a to z důvodu předchozích zkušeností s těmito jazyky. Nakonec však byl zvolen programovací jazyk, který je především kompatibilní s většinou embedded zařízení s operačním systémem a s možností přístupu k internetu. Jedná se o několik webových technologií, které dohromady vytvoří funkční celek s kompletní databází naměřených dat, nastavením zařízení a i s případnou archivací těchto dat. Tyto technologie jsou HTML, PHP, CSS, JavaScript, jQuery a MySQL. Mimo tyto webové technologie byl použit i programovací jazyk C pro propojení webového softwaru s hardwarem.

Dalším úkolem bylo vybrat hardware na, kterém by zmíněné technologie spolehlivě fungovaly, nebyl finančně náročný a byl lehce dostupný a dostatečně výkonný. Pro tuto úlohu byla tedy vybrána velmi oblíbená platforma, a to Raspberry PI 3 popsaná v kapitole (2.9), která splňovala veškeré požadavky.

Celkový řetězec spojený s měřením tlaku je znázorněn na obrázku (17). Avšak tato práce se zabývá pouze webovou aplikací a databází MySQL uvnitř mikropočítače Raspberry Pi.



Obrázek 17: Kompletní měřicí řetězec

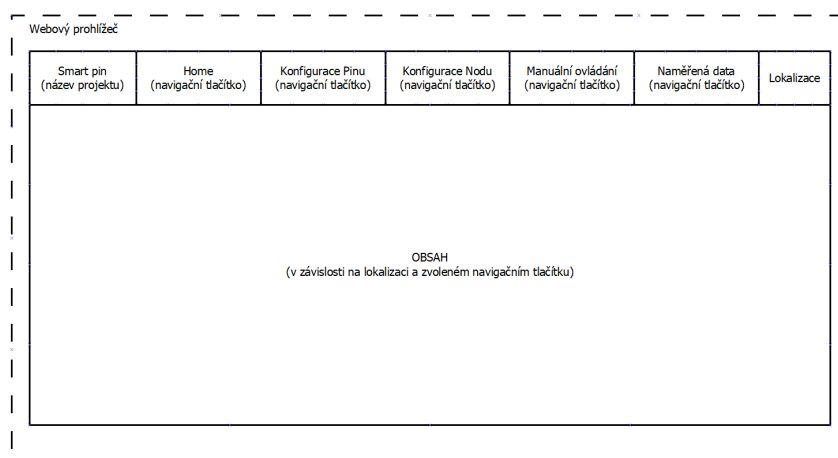
Pro možnost vytvoření webové aplikace byl na Raspberry Pi nainstalován webový server Apache. Dále pro vytvoření databáze byl použit fork MySQL – MariaDB. Databáze je spravována klasickým grafickým rozhraním `phpmyadmin`, které umožňuje činnosti jako například vytváření jednotlivých databází, tabulek, dále také přiřazování práv jednotlivým uživatelům,

a mnoho dalšího. Bližšímu popisu návrhu a vytvoření databáze je věnována kapitola (4.3).

Pro vylepšení grafického vzhledu a taky dynamické přizpůsobivosti rozlišení webové aplikace byla použita sada nástrojů s názvem Bootstrap popsána v kapitole (2.5).

4.1 Návrh webové aplikace

Po několika návrzích konceptu vzhledu webové aplikace pro ovládání zařízení bylo vybráno rozvržení s navigační lištou, která je na horní straně webové stránky. Navigační lišta obsahuje název projektu, odkazy jednotlivých položek v menu a nakonec i lokalizaci do třech různých jazyků (angličtina, němčina a čeština). Pod horní navigační lištou se nachází obsah webu v závislosti na vybrané položce v menu.



Obrázek 18: Koncept webové aplikace

Názvy jednotlivých položek menu jsou:

- home,
- konfigurace pinu,
- konfigurace nodu,
- manuální ovládání,
- naměřená data,
- lokalizace(English – Deutsch – Čeština).

Stránka home

Obsah této stránky je pouze obrázek zobrazující logo firmy.

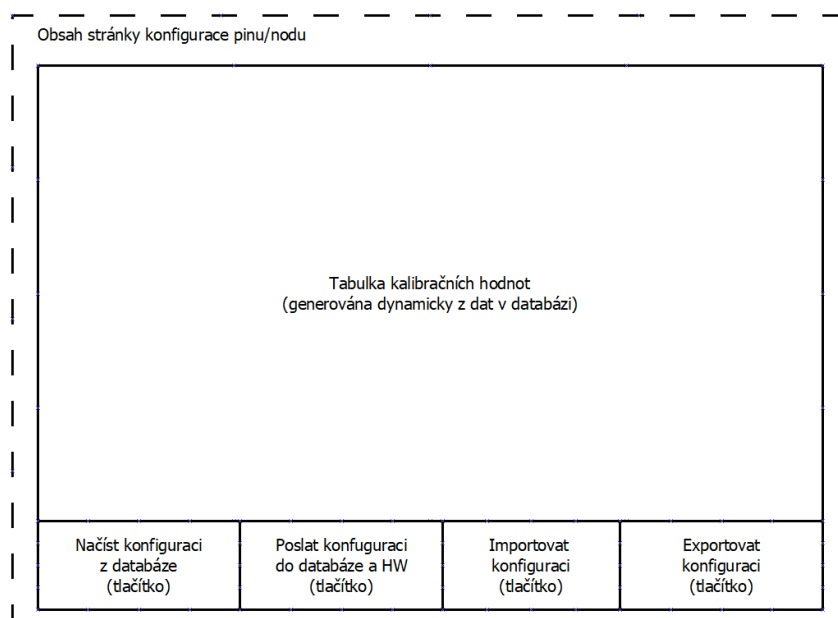
Stránky konfigurace pinu a nodu

Účelem této stránky je zobrazení tabulky, která servisnímu technikovi umožní několik možností,

jako jsou: zobrazit si aktuální nastavené parametry s možností měnit parametry jednotlivých pinů nebo nodů. Další funkce přidává čtveřice tlačítek umístěných pod tabulkou. Tyto funkce jsou:

- načtení konfigurace z databáze,
- zaslání konfigurace do databáze a hardwaru,
- importování konfigurace,
- exportování konfigurace.

Podrobný popis funkčnosti těchto tlačítek je popsán v kapitole (4.2). Graficky znázorněný koncept obsahu stránky pro konfiguraci pinu nebo nodu můžeme vidět na obrázku (19).



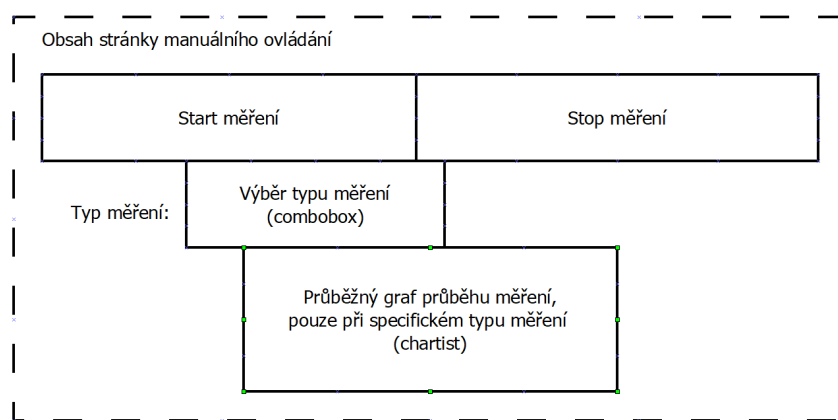
Obrázek 19: Koncept obsahu webové stránky pinu a nodu.

Manuální ovládání

Obsah této stránky má sloužit servisnímu technikovi ke vzdálené možnosti manuálně změřit data. Způsob manuálního ovládání spočívá v zadání typu měření do příslušného **comboboxu**⁸. Poté, co je vybrán typ měření, jsou k dispozici dvě tlačítka. První tlačítko má název **Start měření** a jeho úkolem je zaslat požadavek do hardwaru ke spuštění měření. Druhým tlačítkem **Stop měření** se měření naopak zastavuje. Kromě toho však posílá také požadavek na vrácení dat z daného měření. Data se dále zasílají do databáze. Pokud je vybrán speciální typ měření,

⁸Combobox - Kombinované pole je grafický ovládací prvek umožňující zadávání hodnot pomocí výběru z rozbalovacího seznamu.

který je specifikován v kapitole (4.2), vykreslí se data již v průběhu měření přímo na stránce manuálního ovládání. V ostatních případech se budou moct dát zobrazit na stránce **Naměřená data**. Rozložení této stránky je rozebráno níže. Celkový koncept můžeme vidět na obrázku (20)



Obrázek 20: Koncept obsahu webové stránky manuálního ovládání.

Naměřená data

Obsah této stránky má poskytnout servisnímu technikovi možnost zobrazit si naměřená data jednotlivých měření. A to ať už z měření krátkého nebo dlouhého. Je-li měření krátké, zobrazí se tabulka vybraných hodnot. V druhém případě (u dlouhého měření) je výsledek zobrazen v grafu. Data pro zobrazení vybírá technik pomocí **comboboxu**, ve kterém je nabídka jednotlivých měření ve formátu [číslo měření, datum měření]. Po výběru čísla měření je zapotřebí vybrat pin nebo více pinů, které chceme zobrazit. Pro úplnost tohoto řešení, možnost exportu a další analýzy dat je přidáno tlačítko **Stáhnout zobrazená data**. Celkový koncept obsahu stránky **Naměřená data** je vyobrazen na obrázku (21).

4.2 Implementace jednotlivých částí webové aplikace

V této podkapitole bude řešena realizace zmíněných návrhů do funkčního řešení.

Implementace stránky konfigurace pinu a nodu

K Implementaci tabulky konfigurace byl využit standardní html tag `<table>` doplněn o atribut (**class**) s hodnotami `table`, `table-condensed` a `table-striped`. Hodnota atributu **class** `table` definuje, že se jedná o tabulku. `table-condensed` říká, že budou řádky tabulky užší a poslední třída `table-striped` upraví tabulku tak, že jednotlivé řádky tabulky budou proužkované. V hlavičce tabulky označené tagy `<thead>` jsou odkazy na jednotlivé identifikátory názvu jednotlivých sloupců vycházející ze zvolené lokalizace. Dále v těle tabulky se pod tagy `<tbody>` nachází atribut tagu (`id = live_data`), pomocí kterého je vložen obsah tabulky. Obsah tabulky je vy-

Obsah stránky naměřená data		
Vyberte číslo měření:	číslo měření (combobox)	
tabulka aktivních pinů daného měření (pole checkboxu)		
Graf měření (chartist)		
		stáhnout zobrazené data (tlačítko)

Obrázek 21: Koncept obsahu webové stránky naměřená data.

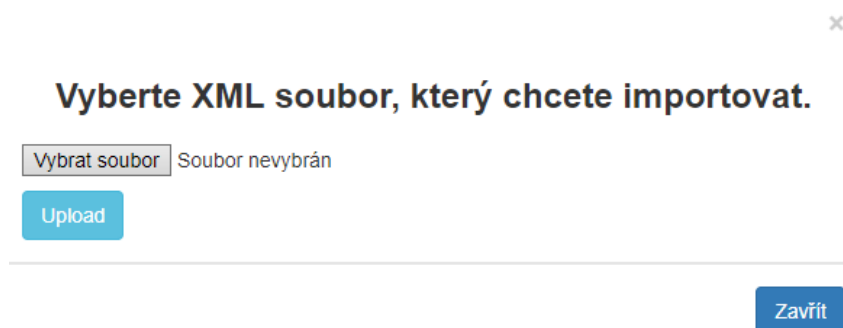
tvořen dynamicky pomocí SQL dotazu na databázi a knihovny AJAX. Jednotlivé data tabulky jsou vloženy do takzvaných `input` boxů (zapisovatelné pole) nebo comboboxů (v závislosti na typu proměnné). V tabulce je použita jQuery funkce `on` s parametrem `blur`, který zavolá funkci pokaždé, když v příslušném políčku tabulky provedeme změny a políčko opustíme. Zavolaná funkce provede uložení čtyř parametrů, a to ID řádku v databázi, adresy pinu, názvu kalibrační konstanty a hodnot, které byly upraveny.

U implementace tlačítka **Načíst konstanty z databáze** je využit tag `<input>`, který obsahuje atributy (*class*) s parametry `btn`, `btn-primary`, `btn-block`. `Btn` určuje, že input bude tlačítko, `btn-primary` zavádí modrou barvu tlačítka a `btn-block` přizpůsobuje velikost prostoru, který mu byl vyhrazen. Dalším atributem je `id`, pomocí kterého se volá knihovní funkce jQuery `click`. Tato funkce provede pouze příkaz `location.reload()`, který obnoví stránku a znova se načtou data z databáze.

Tlačítko konfigurovat má stejný tag a atributy jako předchozí a navíc atribut `id`. Pomocí tohoto tlačítka je vyvolána funkce, která veškeré provedené změny zašle a zapíše do databáze. Tento zápis funguje s knihovnou AJAX, tudíž se vykonává na pozadí. Po dokončení zápisu do databáze automaticky obnoví stránku a tím obnoví tabulku s již nově nastavenými parametry. Kromě změněných parametrů zasílá do jiné tabulky v databázi jménem `command` instrukce o tom, že byla provedena změna konfigurace. Raspberry Pi díky této informaci provede zaslání

konfigurace do příslušného hardware.

Dalším tlačítkem obsahu konfigurace pinu a nodu je **importovat konfiguraci**. Atributy se nijak neliší oproti předchozím tlačítkům. Funkcí tohoto tlačítka je pomocí správně formátovaného *xml* souboru překonfigurovat celou tabulku v databázi **pin** nebo **node** a to z důvodu, aby si servisní technici mohli zálohovat určité typy konfigurací během případného ladění zařízení. Princip chování je ten, že po kliknutí na toto tlačítko je zobrazeno vyskakovací okno **modal**, ve kterém je nabídka **vybrat soubor** a potvrzující tlačítko **upload** pro nahrání *xml* do databáze. Vzhled vyskakovacího okna můžeme vidět na obrázku (22). Složení *xml* souboru pro pin i node je popsáno níže. Po úspěšném importování konfigurace do databáze se stejně, jako u předchozího tlačítka, provede i zápis o změně parametru do tabulky **command**.



Obrázek 22: Okno volby souboru pro importování konfiguračních údajů do databáze.

Posledním tlačítkem je export konfigurace. Tohle tlačítko funguje oproti ostatním jinak, a to tak, že volá PHP dokument, který vyčte obsah dat z databáze v závislosti na tom, zda máme stránku pinu nebo nodu. Po načtení těchto dat je zavolána třída DOM(document object model), která pomocí námi vytvořené struktury tyto data rozdělí a vytvoří *xml* strukturu. Struktura *xml* je tvořena kořenovým elementem `<pins>`. Dále elementem `<pinID = X>` a poté s několika vnitřními elementy obsahujícími údaje obsažené v databázi. Po složení toho *xml* dokumentu je do názvu přidán datum a čas exportování a je uložen do zařízení, ze kterého byl požadavek na export proveden. Vygenerovaný soubor přesně odpovídá typu, který je možné zpětně importovat. Je tudíž možné si tímto způsobem vytvářet zálohy aktuální konfigurace.

Implementace stránky manuálního ovládání

Obsah této stránky je, jak již bylo řečeno dříve, tvořen pomocí dvou tlačítek a výběrového pole combobox.

První z tlačítek pod názvem **Start měření** je tvořeno taktéž pomocí tagu `<input>` s atributy `class` a `id`. Funkce tohoto tlačítka je závislá na výběru typu měření v comboboxu. Tyto typy měření budou popsány níže.

Druhé tlačítko **Stop měření** funguje ve všech typech měření stejně, a to tak, že vloží příkaz do databáze (tabulka `command`). Tento příkaz dále přečte obslužný program a zašle ho třikrát co sto milisekund do hardwaru.

Výběr typu měření je, jak již bylo zmíněno, tvořen *comboboxem*. K vytvoření slouží klasické HTML tagy `<select>` s atributem `class` a hodnotou `form-control`, která pouze vylepšuje vzhled v závislosti na použitém frameworku (v tomto případě bootstrap). Obsah výběru je pevně daný a vložen mezi tagy `<option>`. Tyto typy měření jsou závislé na požadované funkci nebo způsobu měření. Jednotlivé typy měření a jejich účel je popsán níže:

Jako první typ je zmíněn režim, při kterém je požadavek na real time měření (měření v reálném čase). Měření je prováděno pouze na jednom pinu, který je potřeba si zvolit k tomu, aby bylo možné měření spustit a aby byla odemčena možnost stisknutí tlačítka **start měření**. Při splnění všech podmínek pro zahájení měření je po stisku zmíněného tlačítka zaslána zpráva do databáze s příkazem, který zahrnuje daný typ měření a číslo pinu. Na rozdíl od ostatních typů měření má tento typ přímou odezvu na daný pin, který při spuštění měření ihned zasílá data z tlakového senzoru. Tyto data zapisuje do vyhrazeného místa v databázi – konkrétně tabulky `command` a řádku s názvem `data`, kde se kromě příkazu nachází i místo pro data určená k vykreslení průběhu, které se periodicky mění. Webový prohlížeč tyto data periodicky vykresluje v průběžném grafu.

U dalších typů měření je potřeba si ujasnit pojmy **dlouhé měření**, **krátké měření** a způsoby spuštění měření, a to buď napětovou úrovní nebo časem. Dlouhé a krátké měření je vlastně výběr dat, které po dokončení měření dostáváme. Data z krátkého měření jsou pouze základní a neobsahují kompletní informace každého vzorku z průběhu měření. V případě typu dlouhého měření posílají piny celkový průběh od začátku až do konce jednotlivých úseků a tyto úseky jsou určeny způsobem, který bude následně popsán.

Z toho důvodu, že zařízení není nijak spojeno s řídicí jednotkou lisu, bylo potřeba určit, jakým způsobem se bude měření spouštět. Jelikož je výsledné zařízení stále ve vývoji, tak byly specifikovány dva způsoby. První způsob řízený výstupní napětovou úrovní kapacitního senzoru tlaku spouští záznam měření zjištěním náběžné hrany, jejíž úroveň si servisní technik nastavuje pomocí tabulky konfigurace pinu. Taktéž si nastavuje i úroveň, kdy je záznam ukončen a zaslán do databáze. Druhý typ je založen na znalosti časového intervalu měření. Měření začíná se zpožděním po kalibraci a trvá přednastavenou dobu, po které se měření ukončí a zašlou se naměřená

data.

Po vysvětlení jednotlivých pojmů je možné uvést zbývající typy měření, které se liší od prvního tím, že se měření provádí na všech pinech zároveň. Tyto typy jsou v podstatě kombinací dlouhého a krátkého měření se způsobem zachytávání záznamu měřeného průběhu. Tyto typy se nazývají:

- Režim spouštění časem, analýza i histogram
- Režim threshold, analýza
- Režim spouštění časem, analýza
- Režim threshold, analýza i histogram

Implementace stránky naměřených dat

Obsah této stránky, jak již bylo zmíněno, slouží servisnímu technikovi k zobrazení dat jednotlivých měření, které jsou uloženy v databázi. Jednotlivé měření jsou zobrazeny v comboboxu ve tvaru datum, čas a typ (například 2019-03-15 06:06:14 - - - Long). Označení **short** a **long** náleží krátkému a dlouhému měření, které již byly popsány výše. Po výběru měření dále dochází ke dvěma variantám chování v závislosti na typu měření. V prvním případě, pokud je měření typu **short**, je na webové stránce zobrazena tabulka všech pinů se základními daty získanými měřeními. V druhém případě se po výběru dlouhého měření zobrazí nabídka jednotlivých pinů, které v daném měření byly aktivní. Piny jsou zobrazeny pomocí **checkboxů** (zaškrtačích polí), pomocí kterých si servisní technik může vybrat konkrétní piny, ze kterých se následně vykreslí graf celého průběhu měření. Jako poslední funkce u dlouhého typu měření je možnost si data stáhnout do **csv** souboru pro možnost následné analýzy v jiném programu.

Lokalizace

Lokalizace byla vyřešena pomocí PHP *session*, které sdílejí data mezi jednotlivými *http requesty*. Tyto *http requesty* zadávají tlačítka, které mají pod určitým názvem schovanou proměnnou reprezentující daný jazyk. Pomocí této proměnné a příkazů *switch* a *case* je vybrán příslušný jazyk, který pomocí PHP funkce *include* přiřadí danou knihovnu se zvoleným jazykem. Knihovna jazyků je řešena pomocí mnoha *define*, které mají dva parametry – identifikátor a text přiřazený identifikátoru. Identifikátory jsou dále vloženy na místa, kde se má nacházet text pomocí PHP příkazu *echo*. Níže uvedené fragmenty kódu vysvětlují způsob implementace změny jazyka. První z nich (8) je vystřižen z navigačního panelu, kde je možné vidět tři hypertextové odkazy, které zavolají PHP dokument **define_lang.php**. Tento dokument spustí *session*, přijme parametry **lang** a číslo, podle kterých druhý příklad (9) vybere příslušný dokument s daným jazykem. Fragment kódu dokumentu českého jazyka je znázorněn na třetím příkladu (10), kde, jak již

bylo zmíněno, je přiřazen text pod určitý identifikátor.

```
<ul class="nav navbar-nav navbar-right">
  <li><a href="languages/switch_lang.php?lang=1">English</a></li>
  <li><a href="languages/switch_lang.php?lang=2">Deutsch</a></li>
  <li><a href="languages/switch_lang.php?lang=3">Čeština</a></li>
</ul>
```

Výpis 8: Ukázka části kódu navigační lišty webové aplikace

```
<?php
session_start();
if (!isset($_SESSION['lang'])) {
    $_SESSION['lang'] = 1;
}
switch ($_SESSION['lang']) {
    case 3:
        include('languages/lang_cs.php');
        break;
    case 2:
        include('languages/lang_de.php');
        break;
    default:
        include('languages/lang_en.php');
        break;
}
?>
```

Výpis 9: Ukázka souboru define_lang.php

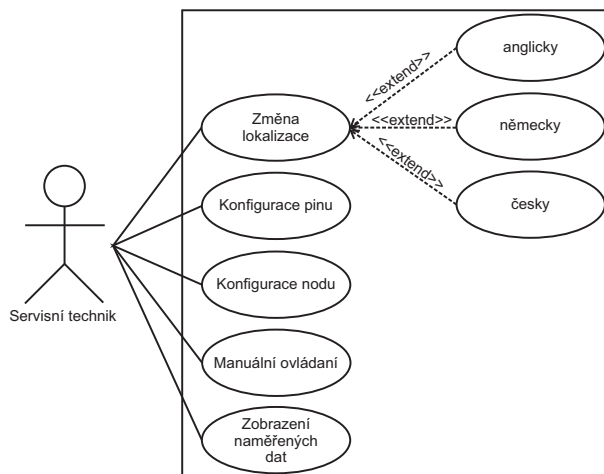
```
<?php

define("_t_title","Smart pin");//nazev webu
define("_t_home","Home");
define("_t_pin_conf","Konfigurace kolíku");//konfigurace kolíku
define("_t_node_conf","Konfigurace Nódu");//konfigurace node
define("_t_man_con","Manualní ovládání");//manualni ovladani
define("_t_meas_data","Naměřená data");//naměřena data
?>
```

Výpis 10: Část výpisu kódu ze souboru lang_cs.php

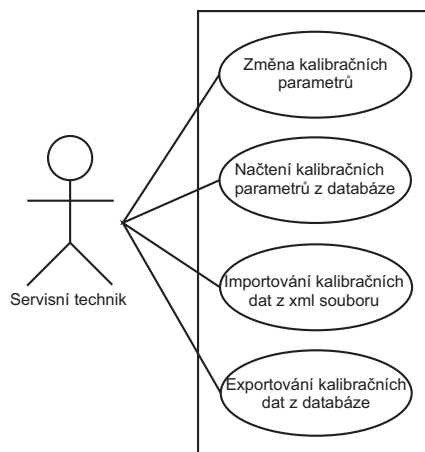
4.2.1 Shrnutí

V této kapitole se nachází shrnutí funkcí pomocí **use case diagramů**, které webová aplikace nabízí servisnímu technikovi. První věc, kterou webová aplikace nabízí, jak můžeme vidět na diagramu (23), je kompletní změna jazyka na všech stránkách webové aplikace. Je možné volit ze tří jazyků (angličtina, němčina a čeština). Možnosti jednotlivých stránek jsou shrnuty na následujících diagramech případů užití.



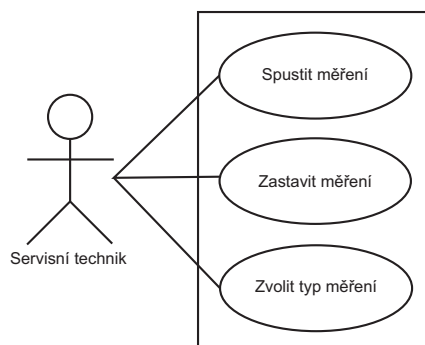
Obrázek 23: Případy užití webové aplikace.

V případě dalšího diagramu (24), který ukazuje, jaké možnosti se nabízí v obsahu stránky konfigurace, ať už pinu nebo nodu, můžeme vidět, že je možné měnit parametry konfigurace jednotlivých zařízení. Další možností je v podstatě obnovení tabulky tím, že se opětovně načte z databáze. To slouží k především k zjištění, zda veškeré piny a nody komunikují, ale také i ke smazání nechtěných a nezaslaných změn v tabulce konfigurace. Dále je servisnímu technikovi poskytnuta jak možnost si celkovou konfiguraci uložit do vlastního zařízení, tak i možnost tuto konfiguraci zpětně nahrát do databáze a tím i do koncového hardwaru pinu nebo nodu.



Obrázek 24: Případy užití konfigurace pinu/nodu.

Jak je znázorněno v diagramu (25), může být servisnímu technikovi poskytnuto manuální ovládání. Může tak v podstatě ovládat celý měřicí řetězec, a to tím, že si nastaví potřebné typy měření. Například měření na jednom pinu nebo způsob měření na všech pinech podrobně popsaných výše. Následně může měření spustit nebo zastavit probíhající měření.

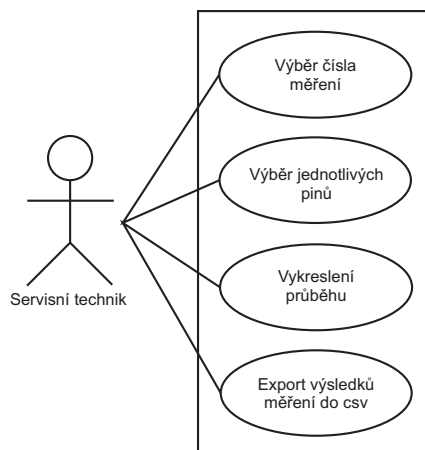


Obrázek 25: Případy užití manuálního ovládání.

V posledním diagramu (26) znázorňujícím možnosti webového obsahu stránky **naměřená data** můžeme vidět, že servisní technik si může zvolit libovolné měření, které bylo uloženo v databázi. Dále si může zvolit jednotlivé piny, na kterých se měření vykonávalo. Data se ihned po výběru jednoho nebo více pinů vykreslí do jednoho grafu. Zde může servisní technik ihned poznat, zda měření nezaznamenalo nějakou poruchu, a následně si zobrazená data vyexportovat do svého zařízení pro možnosti další analýzy.

4.3 Návrh MySQL databáze

Základem projektu **Smart cushion pin** je měření deformace lisovacích válců a následné vyhodnocení tlaků, které ve válcích probíhaly. Jelikož je měřeno několik válců zároveň, byla potřeba



Obrázek 26: Případy užití naměřených dat.

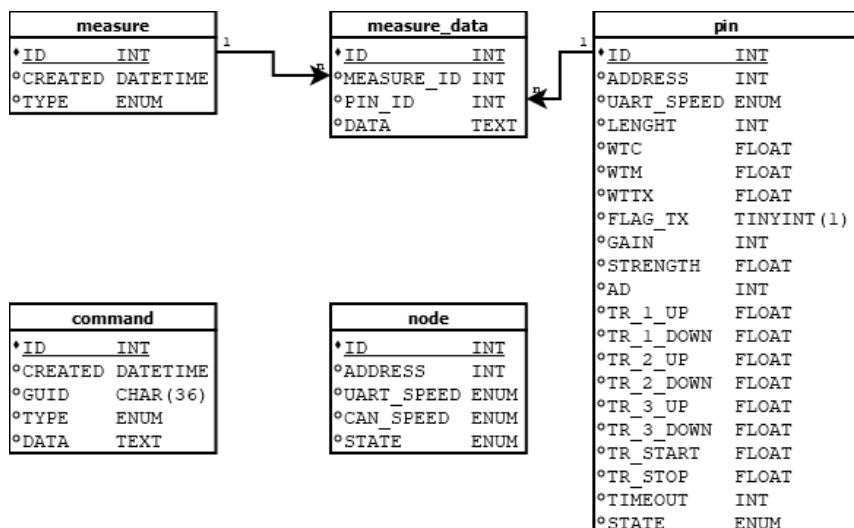
tyto data z měření někde ukládat. Dále byla potřeba ukládat i potřebná kalibrační data, které byly potřeba například ke správnému přepočtu hodnoty senzoru na výsledný tlak. K tomuto účelu byla vybrána relační databáze, která nabízela již přichystaný systém pro zpracovávání objemných dat. Pro systém řízení báze dat byla proto vybrána relační databáze MySQL od firmy MySQL AB, která je volně dostupná a pro potřeby tohoto projektu s celá dostačující.

Aby bylo možné MySQL provozovat, bylo potřeba vytvořit databázový server, který podporuje právě MySQL databáze. Tento server se nazývá MariaDB a běží na lokálním zařízení. Jako vylepšení byla kromě databázového a PHP severu na Raspberry Pi doinstalována i webová aplikace `phpmyadmin` vytvořená pro správu databáze bez nutnosti používat pouze příkazový řádek.

Po zprovoznění potřebných technologií pro chod databázového systému byla potřeba vymyslet strukturu odpovídající požadavkům projektu. Tato struktura byla vytvořena použitím pěti tabulek v databázi. Celková struktura databáze je zobrazena na obrázku (27). Následně budou níže popsány jednotlivé tabulky spojeny s touto strukturou. Nejprve je třeba dodat, že názvy tabulek jsou tvořeny malými písmeny. Jednotlivé názvy řádků jsou psané velkým písmem pro větší přehlednost a zabránění chybám kódu – Raspberry Pi má operační systém Linux, jehož výchozí souborové systémy rozlišují velikost písma v názvech.

První tabulka (1) s názvem `measure` slouží pro uchování základních dat, a to časové známky vložené do řádku pod názvem `CREATED`, který má datový typ `datetime`. Dále řádek `TYPE`, do kterého je vložen údaj, zda bylo měření krátké nebo dlouhé.

Tabulka (2) `measure data` obsahující data jednotlivých pinů je tvořena dvěma cizími klíči propojujícími tuto tabulku s tabulkami `pin` a `measure`. Tomuto propojení se říká `foreign key constraint`. Toto propojení utváří podmínky, které jsou potřebné k tomu, aby do databáze



Obrázek 27: Celková struktura databáze.

Tabulka 1: (measure) číslo měření.

Zkratka	Popis	Rozsah a jednotka
CREATED	Datum a čas měření	
TYPE	Typ měření	long/short

mohly být vloženy data. Určují například, že u tabulky **measure_data** nemůže existovat záznam měření pro ID měření a pin, které by v tabulkách **measure** a **pin** neexistovaly. Posledním atributem jsou **DATA**.

Obsah atributu **DATA** je závislý na informaci z tabulky **measure**. Konkrétně na atributu **TYPE**, který říká, jaký typ měření byl proveden. V podstatě je rozdíl akorát v textovém řetězci, který je v atributu **data** uložen. Oba řetězce obsahují nejprve parametry, při kterých bylo měření provedeno. Dále je uloženo pár statistických údajů. Poslední část se týká pouze dlouhého měření a to tak, že jsou na konci tohoto řetězce uloženy všechny hodnoty vzorků z celého záznamu měření. Stručný popis řetězce pro ukládání dat z měření je popsán pomocí tabulky (6).

Tabulky konfigurace **pinu** (3) a **nodu** (4) popisují jednotlivé parametry potřebné jak pro účely fungování měřicího zařízení, tak i pro konfiguraci samotného měření. Obě tyto tabulky obsahují informace ke každému pinu nebo nodu, jako například **ADDRESS**, která je specifická pro každý **pin** a **node**. Dalšími parametry jsou například rychlost komunikace, parametry pro měření a v poslední řadě také informace o stavu jednotlivých **pinů** a **nodů**. **STATE** může nabývat tří stavů, a to **OK** – funguje, **NOK** – nefunguje a **Unknown** – neznámý stav.

Poslední tabulkou **command** (5) je v podstatě řízeno veškeré dění mezi webovou aplikací a hardwarem. Pomocí této tabulky jsou zadávány instrukce pro aktivaci různých druhů měření

Tabulka 2: (measure_data) data měření.

Zkratka	Popis	Rozsah a jednotka
MEASURE_ID	Constraint na číslo měření (ID)	
PIN_ID	Constraint na adresu pinu	
DATA	Typ měření	Short: [GAIN, STRENGTH, AD, TRstart, TRstop, TR1up, TR1down, TR2up, TR2down, TR3up, TR3down, MAX, dt_start_stop, dt1, dt2, dt3, Sum_start_stop, Sum1, Sum2, Sum3] Long: [GAIN, STRENGTH, AD, TRstart, TRstop, TR1up, TR1down, TR2up, TR2down, TR3up, TR3down, MAX, dt_start_stop, dt1, dt2, dt3, Sum_start_stop, Sum1, Sum2, Sum3, Histogram_data_array,...]

Tabulka 3: (pin) parametry pinu.

Zkratka	Popis	Rozsah a jednotka
ADDRES	Adresa pinu	1 ... 120
UART_SPEED	Rychlost UARTu	1200; 2400; 4800; 9600 Baud
LENGTH	Délka měření	1 - 65535
WTC	Čas kalibrace	0 - 25,5s
WTM	Čekání před zahájením měření	0 - 25,5s
WTTX	Čekání na odpověď	1 - 655,35s
FLAG_TX	Příznak odesílání	True/False
GAIN	Rozsah měření	100 - 50000
STRENGTH	Síla	1 - 1100 kN
AD	Hodnota ADC převodníku	0 - 255
TR_1_UP	Treshold nástupné hrany	0 - 100 %
TR_1_DOWN	Treshold sestupné hrany	0 - 100 %
TR_2_UP	Treshold nástupné hrany	0 - 100 %
TR_2_DOWN	Treshold sestupné hrany	0 - 100 %
TR_3_UP	Treshold nástupné hrany	0 - 100 %
TR_3_DOWN	Treshold sestupné hrany	0 - 100 %
TR_START	Treshold začátku měření	0 - 100 %
TR_STOP	Treshold konce měření	0 - 100 %
TIMEOUT	Maximální doba měření	1 - 255 s
STATE	Stav zařízení	OK; NOK; Unknown

Tabulka 4: (node) parametry nodu.

Zkratka	Popis	Rozsah a jednotka
ADDRES	Adresa nodu	1 ... 63
UART_SPEED	Rychlost UARTu	1200; 2400; 4800; 9600 Baud
CAN_SPEED	Rychlost CANu	125; 250; 500 kbps
STATE	Stav zařízení	OK; NOK; Unknown

Tabulka 5: (command) příkazy.

Zkratka	Popis
CREATED	Datum a čas zaslání příkazu
GUID	Unikátní identifikátor příkazu
TYPE	Žádost / Odezva
DATA	Číslo příkazu

Tabulka 6: Popis jednotlivých zkratk v textovém řetězci DATA.

Zkratka	Popis
GAIN	Rozsah měření
STRENGTH	Síla
AD	Hodnota ADC převodníku
TRstart	Threshold začátek měření
TRstop	Threshold konec měření
TR1up	Threshold začátek měření sekce 1
TR1down	Threshold konec měření sekce 1
TR2up	Threshold začátek měření sekce 2
TR2down	Threshold konec měření sekce 2
TR3up	Threshold začátek měření sekce 3
TR3down	Threshold konec měření sekce 3
MAX	Maximální hodnota tlaku měřené křivky
dt_start_stop	Vzorky měřené křivky
dt1	Vzorky na sekci 1
dt2	Vzorky na sekci 2
dt3	Vzorky na sekci 3
Sum_start_stop	Integrál měřené křivky
Sum1	Integrál měřené sekce 1
Sum2	Integrál měřené sekce 2
Sum3	Integrál měřené sekce 3
Histogram_data_array	Naměřená data

nebo požadavku na hardware. Princip je takový, že webová stránka vytvoří `command` (příkaz) s parametry datum, identifikátor, typ a číslo příkazu. Obslužný program na Raspberry Pi periodicky kontroluje, zda je nějaký příkaz vytvořen, a pokud ano, tak podle čísla příkazu provede potřebné kroky k vykonání jeho účelu.

5 Testování vytvořeného řešení

Vývoj webových stránek je poměrně náročná činnost a celkový proces od návrhu po finální verzi stojí mnoho času a úsilí. Testování patří mezi velmi důležitou část vývoje webových stránek a neměla by se zanedbat, neboť poté hrozí nežádoucí chování těchto stránek. Uplatněné postupy testování a v některých případech i ošetření nežádoucích stavů bude popsáno v této kapitole.

5.1 Technické parametry vytvořené webové stránky

V rámci testování bylo zahrnuto několik faktorů webové stránky, jako je třeba čas načítání jednotlivých stránek a celková velikost stažených dat z webu. Testování bylo prováděno na dvou různých webových prohlížečích, a to Opera a Google Chrome. Prvním krokem bylo testování rychlosti zobrazení jednotlivých stránek. Způsob testování probíhal pomocí ladícího nástroje, který je obsažen jak v prohlížeči Opera, tak v Google Chrome.

Každá webová stránka byla pomocí klávesové zkratky **Ctrl + R** pětikrát obnovena a čas této obnovy byl zapsán a vyhodnocen v tabulce (7). Pomocí tohoto způsobu je obnovena jen část webové aplikace neboť některé soubory jsou uchovány v mezipaměti. Proto byl proveden i druhý test s použitím klávesové zkratky **Ctrl + F5**, který nahravá kompletní webovou stránku bez použití mezipaměti. Časy v druhém testu se téměř zdvojnásobily a shrnutí můžeme opět vidět v tabulce (8). Z obou testů je na první pohled vidět, že prohlížeč Google Chrome má nižší odezvu.

Velikosti stažených dat při načtení webové stránky se v závislosti na prohlížeči nijak nelišily a u všech čtyř testovaných stránek měly téměř totožnou hodnotu, která se pohybovala v rozmezí 522 až 529 kB. Celková velikost všech souborů webové stránky uložené na disku činí 1,2 MB

5.2 Ukázky ošetření nežádoucích stavů

Tabulka konfigurace

Při testování tabulky docházelo v případě změny stejného parametru dvakrát nebo vícekrát, že se změny uložily pokaždé a při následném potvrzení tlačítkem **konfigurovat** se zbytečně nastavoval stejný parametr v databázi vícekrát. Z toho důvodu byla vytvořena funkce, která hledá, zda nebyl parametr již změněn. Pokud ano, tak nahradí předchozí hodnotu hodnotou novou. Tato funkce se nazývá **add_into_buffer** a obsahuje tři vstupní parametry: id řádku, název sloupce a nastavený parametr. Je složena ze dvou cyklů *for*. První z nich zjišťuje, zda nová změna není na stejném řádku jako již dříve provedené změny. Druhý cyklus zjišťuje, zda není shoda s názvem sloupce. Pokud tuto shodu najde, tak je použit příkaz *splice*, který nahradí starý parametr za nový. Celou funkci můžeme vidět ve výpise (11).

Tabulka 7: Časy načítání webové stránky po stisku **Ctrl + R**

	Prohlížeč					
	Opera			Google Chrome		
	Čas v ms					
Stránka	MIN	MAX	AVG	MIN	MAX	AVG
Konfigurace pinu	229	262	244	144	178	162
Konfigurace nodu	223	271	239	140	163	147
Manualní ovládání	225	298	252	137	159	148
Naměřená data	228	258	242	134	153	147

Tabulka 8: Časy načítání webové stránky po stisku **Ctrl + F5**

	Prohlížeč					
	Opera			Google Chrome		
	Čas v ms					
Stránka	MIN	MAX	AVG	MIN	MAX	AVG
Konfigurace pinu	349	392	362	209	224	217
Konfigurace nodu	320	357	342	208	249	233
Manualní ovládaní	342	413	370	211	255	227
Naměřená data	305	348	323	208	248	221

```

function add_into_buffer(id_pin,column,param){
  for (var x = 0;x < id_pin_b.length;x++){
    if (id_pin_b[x] == id_pin){
      for(var y = 0;y < column.length;y++){
        if (column_pin_b[y] == column && y == x){

          id_pin_b.splice(y,1);
          column_pin_b.splice(y,1);
          param_pin_b.splice(y,1);
        }
      }
    }
  }
  id_pin_b.push(id_pin);
  column_pin_b.push(column);
  param_pin_b.push(param);
}

```

Výpis 11: Ošetření stejné volby.

Tlačítko konfigurovat

U tohoto tlačítka je z důvodu větší přehlednosti a také k zamezení nechtěné změny parametru dodělána funkce, která pomocí vyskakovacího okna ukáže, jaké parametry byly změněny. Okno nabídne možnost zavřít (a nezaslat novou konfiguraci do databáze) a možnost potvrdit změny a zaslat konfiguraci do databáze a hardwaru.



Obrázek 28: Ukázka okna změn a volby možnosti zápisu do databáze a hardwaru.

Tlačítko vyčíst konfiguraci

U tohoto tlačítka bylo zjištěno, že neúmyslné kliknutí vedlo k obnovení tabulky a tím pádem ke ztrátě provedených změn. Z tohoto důvodu bylo přidáno vyskakovací okno s dotazem, zda opravdu chceme tabulku obnovit. Vyskakovací okno je vyvoláno pouze v případě, že byly provedeny nějaké úpravy v tabulce. V Opačném případě dojde k pouze k obnovení stránky a tím i tabulky.

6 Závěr

Diplomová práce se zabývala vytvořením webových stránek, které umožňují konfiguraci a vizualizaci pro zařízení měřící tlak. Mimo to bylo cílem seznámit se s použitými technologiemi. Práce byla rozdělena do několika kapitol.

Kapitola (2) sloužila k seznámení se s použitými technologiemi pro vývoj moderní webové aplikace. A to jak s technologiemi ovlivňujícími styl takové stránky, tak jeho chování. Mimo webové technologie nás tato kapitola měla seznámit s použitou platformou Raspberry Pi. Byl podrobně popsán jak vývoj této platformy, tak i obsažený hardware.

Následující kapitola (3) sloužila k seznámení s technologií, která se používá k měření tlaku. Konkrétně s kapacitním senzorem tlaku, který vyniká svou přesností a patří také mezi jedny z nejpoužívanějších senzorů.

V kapitole (4) byl popsán jak kompletní návrh rozložení jednotlivých komponent webové aplikace, tak i její implementace s použitím frameworku Bootstrap pro moderní vzhled, JavaScriptu pro dynamické chování a PHP pro možnosti komunikace s databází. Mimo návrhu a implementace webové aplikace byl řešen i návrh samotné databáze v MySQL pro uchovávání jak naměřených dat, tak i konfiguračních parametrů zařízení pro měření tlaku.

V poslední kapitole (5), která se zabývala testováním vytvořeného řešení, bylo provedeno několik testů webové stránky. Testy se týkaly rychlosti načítání nebo velikosti stažených dat u různých typů webových prohlížečů.

Na závěr bych rád řekl, že tato práce je součástí většího projektu, který pokračuje i po odevzdání této diplomové práce. Tudíž se na vývoji webové aplikace bude dále pokračovat a v této práci byla probírána pouze průběžná verze webové aplikace.

Literatura

- [1] HOVORKA, Karel. *KIVFS - Webový klient* [online]. Plzeň, 2013 [cit. 2019-04-16]. Dostupné z: <<https://theses.cz/id/nf3abt/>>. Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd. Vedoucí práce Ing. Luboš Matějka.
- [2] *Besant Technologies* [online]. 2017 [cit. 2019-04-09]. Dostupné z: <https://www.besanttechnologies.com/difference-html-html5-css-css3>
- [3] TRAN, An. *Návrh a realizace sesterské aplikace pro RTLS*[PDF]. Ostrava, 2018 [cit. 2019-04-16]. Diplomová práce. Technická univerzita Ostrava Fakulta elektrotechniky a informatiky Katedra kybernetiky a biomedicínského inženýrství. Vedoucí práce Ing. Jaromír Konečný, Ph.D.
- [4] ČÁPKA, David. *Objektově orientovaného programování* [online]. [cit. 2019-04-09]. Dostupné z: <https://www.itnetwork.cz/csharp/oop/c-sharp-tutorial-uvod-do-objektove-orientovaneho-programovani>
- [5] *PHP* [online]. [cit. 2019-04-09]. Dostupné z: <https://www.php.net/manual/en/history.php.php>
- [6] *Úvod do CSS frameworku Bootstrap* [online]. [cit. 2019-04-09]. Dostupné z: <https://www.itnetwork.cz/html-css/bootstrap/kurz/uvod-do-css-frameworku-bootstrap>
- [7] *Bootstrap History* [online]. [cit. 2019-04-09]. Dostupné z: <https://v4-alpha.getbootstrap.com/about/history/>
- [8] *Learning Bootstrap*. Packt Publishing, 2014. ISBN 978-1-78216-184-4.
- [9] *Pinclipart* [online]. [cit. 2019-04-09]. Dostupné z: https://www.pinclipart.com/pindetail/ihxmmT_bootstrap-bootstrap-4-logo-png-clipart/
- [10] *JQuery návod* [online]. [cit. 2019-04-09]. Dostupné z: <http://jquery-navod.cz/kategorie-ostatni-clanky/1-uvodni-clanek>
- [11] *JQuery pro neprogramátory*. ALBATROS MEDIA a.s. Na Pankráci 30, Praha 4: Computer Press, 2012. ISBN 978-80-251-3750-5.
- [12] *Skillcrush* [online]. [cit. 2019-04-09]. Dostupné z: <https://skillcrush.com/2018/04/26/what-is-ajax/>
- [13] *Data Science Central* [online]. [cit. 2019-04-09]. Dostupné z: <https://www.datasciencecentral.com/profiles/blogs/history-of-mysql>
- [14] FITZPATRICK, JASON. *Raspberry Pi* [online]. [cit. 2019-04-09]. Dostupné z: <https://www.howtogeek.com/138281/the-htg-guide-to-getting-started-with-raspberry-pi/>

- [15] *Raspberry Pi* [online]. [cit. 2019-04-09]. Dostupné z:
<https://www.raspberrypi.org/documentation/usage/gpio/>
- [16] *Reichelt elektronik* [online]. [cit. 2019-04-09]. Dostupné z:
<https://www.reichelt.com/de/en/raspberry-pi-3-b-4x-1-4-ghz-1-gb-ram-wlan-bt-raspberry-pi-3b-p217696.html?r=1>
- [17] *Smar* [online]. [cit. 2019-04-09]. Dostupné z: <http://www.smar.com/en/technical-article/pressure-measurement-characteristics-technologies-and-trends>
- [18] *Suga* [online]. [cit. 2019-04-09]. Dostupné z: <http://cz.sugameasure.com/pressure-gauge/wika-pressure-gauge/bourdon-tube-wika-gas-pressure-gauge.html>
- [19] SCHWARZ, Jakub. *Kalibrace tlakoměrů* [online]. Brno, 2014 [cit. 2019-04-16]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace?zp_id=68884. Diplomová práce. Fakulta strojního inženýrství, Aplikovaná informatika a řízení. Vedoucí práce Ing. František Vdoleček, CSc.
- [20] *Eastsensor* [online]. [cit. 2019-04-09].
Dostupné z: <https://www.eastsensor.com/blog/capacitance-differential-pressure-transmitter-working-principle/>
- [21] *Metropolia* [online]. [cit. 2019-04-09].
Dostupné z: <https://wiki.metropolia.fi/display/sensor/Pressure+sensor>